Cover Page – MSc Business Analytics Consultancy Project/Dissertation 2020-21

Title of Project: ***Learning Data-Driven Insights Into the Incidence of Homelessness in the UK Using Text-based Clustering of Crisis Helpline Records***

Date: September 1st, 2021

Word Count: 11,153

Disclaimer:

I hereby declare that this dissertation is my individual work and to the best of my knowledge and confidence, it has not already been accepted in substance for the award of any other degree and is not concurrently submitted in candidature for any degree. It is the end product of my own independent study except where other acknowledgement has been stated in the text.

Abstract

This dissertation explores and evaluates the performance of various feature extraction algorithms engaged in unsupervised clustering tasks with free text data. The data employed in this research was provided by Centrepoint, a UK-based charity focused on ending youth homelessness. The directive of this project was to analyze data from the Centrepoint crisis helpline to identify segments of records sharing similar characteristics. Initial data exploration revealed most available categorical features were not suitable for Machine Learning due to the presence of null values, leading to a focus on Natural Language Processing. A variety of free text feature extraction algorithms were experimented on with pre-processed free text data, and the resulting features were employed to train an unsupervised k-means clustering model. Several limitations, including document length and complications surrounding the data collection process contributed to inconsistent performance from the entire of set feature extraction approaches. However, the key insight of this analysis is that the vectorization and sentence encoding feature extraction methods performed better than the more advanced deep transfer learning models, which provides insight into the data set characteristics most suited to various feature extraction frameworks. After each feature extraction method was tested, the evaluation criteria determined that a k-means clustering model fitted with features extracted by the TF-IDF model, run on the helpline data set produced the most robust results. This model produced ten distinct clusters and revealed that housing situation features produced the greatest share of inter-cluster variation. The distinctive characteristics of each cluster provided insight into current and prospective housing situations, reasons for contacting Centrepoint, and the most pressing current needs of these young people.

Acknowledgments

I would like to acknowledge the support I received throughout this process from my family, UCL faculty, and Centrepoint staff. I would not have been able to complete this dissertation nor earn my degree without your guidance and encouragement along the way. I would specifically like to thank my academic advisor Mahmoud Elbattah for refining my research methods and helping shape the direction of this project, Jennifer Barnes and Daniel Poursaeedi for all their support from Centrepoint, and Susannah Keys for making my writing look much better than it is.

Table of Contents

List of Figures……..……..……...7
List of Tables……..……..……..……...8
List of Acronyms……..……..……..……...9

1. Introduction……..……..……...10
1.1 Consulting Case……..……..……10
1.2 Organizational Objective……..……..……..10
1.3 Technical Approach……..……..……..……11

2. Technical Background……..……..……..……..12
2.1 Feature Extraction from Natural Language ……..……12
2.1.1 Vectorization……...……..……....12
2.1.2 Word Embedding……..……....13
2.1.3 Deep Neural Networks……..……..…..14
2.2 Clustering Methods……..……..……....17
2.2.1 Centroid-Based Clustering……..……..17
2.2.2 Gaussian Mixture Models……..……..……...18
2.2.3 Hierarchical Clustering……..……...…….18
2.2.4 Density-Based Clustering……..…..…..……19

3. Literature Review……...……..20
3.1 Related Works in Natural Language Processing……..……..…....21
3.1.1 Configuring CNNs for NLP……..……..21
3.1.2 Transformer Models……..…....21
3.1.3 Short Text Clustering……..……....23
3.2 Related Works in Public Health.……..…....23

4. Methodology……..……25
4.1 Overview of Data Sets……..…..……...25
4.2 Data Quality Issues…..……..…..……...25
4.3 Exploratory Data Analysis…..……..……26
4.4 Proposed Methodology……..……..……..30
4.5 Data Pre-Processing……..……..……..…32
4.5.1 Removing Stop Words, Symbols, and Undesired Characters…..…..32
4.5.2 Spell Checking……..……..……..…….33
4.5.3 Tokenization……...……..……..33
4.5.4 Stemming and Lemmatization……...……..……..34
4.6 Free Text Feature Extraction Algorithms Selection……...……..35
4.6.1 Term Frequency - Inverse Document Frequency (TF-IDF)…...…...35
4.6.2 Paragraph Vector (Doc2Vec)……..……...……..…..…37
4.6.3 Paraphrase Distill RoBERTa and MP-Net……....…37

4.7 Unsupervised K-means Clustering……..……..…..38
4.7.1 Mini-Batch and Accelerated K-means……..…...……..38
4.7.2 Hyperparameter Tuning……..……..….38
4.8 Evaluation Metrics…….……..….39
4.8.1 Elbow Method……...……..…..39
4.8.2 Silhouette Score……..…….……..…40
4.8.3 Davies Bouldin and Calinski Harabasz Indices……....…40

5. Results…….…...……...41
5.1 Model Selection……...…….……41
5.2 Clustering Analysis of Best Performing Model…….……...…44
5.3 Organizational Implications……..…….……..45
5.4 Limitations……...…….……..….46

6. Conclusions and Future Work……..……...……...46

Appendices…….……...48
Appendix A: Links to Project Code and Resource Board…….……48
Appendix B: Cluster Word Clouds for TF-IDF Model…….…….…..……49

Bibliography…….……..……...…….……..50

List of Figures

Figure 1: Analysis Process Flow……12
Figure 2: Word Embedding Diagram……14
Figure 3: DNN Diagram……15
Figure 4: RNN Diagram……16
Figure 5: CNN Diagram……17
Figure 6: Non-Spherical Clustering (GMM)……18
Figure 7: Spherical Clustering (K-Means)……18
Figure 8: Counts by Gender……27
Figure 9: Age Distribution……27
Figure 10: Age Distribution by Gender……27
Figure 11: Counts by Region……27
Figure 12: Top 10 Housing Situations……28
Figure 13: Top 10 Reasons for Contacting the Helpline……28
Figure 14: Helpline Word Cloud……29
Figure 15: Enquiry Word Cloud……29
Figure 16: Helpline Average Document Length……29
Figure 17: Enquiry Average Document Length……29
Figure 18: Helpline Most Common Bigrams……30
Figure 19: Enquiry Most Common Bigrams……30
Figure 20: TF-IDF Equation……36
Figure 21: Elbow Method Example for K-Means……40
Figure 22: Elbow Method Chart for TF-IDF Algorithm on Helpline Data……41
Figure 23: Silhouette Score Charts for TF-IDF Algorithm on Helpline Data……42

List of Tables

Table 1: Literature Review Search Parameters……..……..20

Table 2: Data Set Summary……...…….…….……...25

Table 3: Usable Categorical Features in the Helpline Data Set……...26

Table 4: Pre-Processed Free Text Features……..…….……..35

Table 5: Model Evaluation Metric Comparison……..43

Table 6: Count of Records by Cluster……...……..44

Table 7: Descriptive Cluster Analysis……...…….45

List of Acronyms

ML – Machine Learning

NLP – Natural Language Processing

TF-IDF – Term Frequency- Inverse Document Frequency

PCA – Principal Component Analysis

DNN – Deep Neural Network

RNN – Recurrent Neural Network

CNN – Convolutional Neural Network

DBSCAN – Density-Based Spatial Clustering of Applications with Noise

DCNN – Dynamic Convolutional Neural Network

BERT – Bidirectional Encoder Representations from Transformers

RoBERTa – Robustly Optimized BERT Pre-Training Approach

NSP – Next Sentence Prediction

STC – Short Text Clustering

MLP – Multi-Layer Perceptron

OOV – Out-of-Vocabulary

PV-DM – Paragraph Vector-Distributed Memory

PV-DBOW – Paragraph Vector-Distributed Bag-of-Words

SBERT – Siamese Bidirectional Encoder Representations from Transformers

SGD – Stochastic Gradient Descent

DB Index – Davies Bouldin Index

CH Index – Calinski Harabasz Index

1. Introduction

1.1 Consulting Case

This project was conducted in partnership with Centrepoint[1], a charity focusing on ending youth homelessness in the UK. Centrepoint provides housing, healthcare services, and career and life-skill training programs to people aged 16-24 who are experiencing or are at risk of experiencing homelessness. Each year, Centrepoint's crisis helpline receives thousands of queries from young people and youth advocates (such as family members, social workers, and others) in search of assistance with unstable housing situations. Centrepoint helpline representatives field these enquiries by phone, web chat, and email, and record relevant case details, including demographic data, geographic location, and current housing situation. Due to the sensitivity of the data, all records were anonymized for the purposes of this project.

This dissertation is structured as to provide a comprehensive review of the analysis process. It begins by stating the business case and proposed technical approach, elaborating on relevant technical background information and related works, and contextualizing the available data. This is proceeded by an outline of the research methodology, and finally a summary the results, key findings, and organization implications of this research. Links to the Jupyter notebook containing the code for this analysis and the project management board used to track its progress can be found in Appendix A.

1.2 Organizational Objective

This project endeavors to use the helpline data to discover distinct clusters of young people with similar characteristics. Currently, Centrepoint catalogues these records and uses the data to help determine the best course of action in individual cases and improve the living circumstances of the young people they represent. However, their ability to identify broader trends in is limited as many of their resources are devoted to tailoring individualized approaches for each case they take on. This is especially the case with free text data, as the organization stores detailed notes of each conversation, but does not currently have the resources to conduct analyses of similar recurring themes and topics among the various groups of young people they serve.

[1] For more information on Centrepoint, please refer to their website at https://centrepoint.org.uk/

Therefore, the goal of this analysis is to use the entirety of Centrepoint's crisis helpline data, with a focus on free text features, to facilitate a more nuanced understanding of the data than is possible with descriptive analytical methods. Insights gleaned from this work may allow Centrepoint to match their services or marketing efforts more efficiently to the specific needs of each subset of young people they serve. In preliminary project discussions, the Centrepoint team provided examples of the type of groupings they anticipated finding, which included concentrations of specific priority need groups in certain locations, varying demographic trends across causes of homelessness, and variation in the current housing situation of young people.

1.3 Technical Approach

The data structure and orientation of the organizational problem lend themselves to unsupervised clustering algorithms. Unsupervised Machine Learning (ML) models derive patterns from unlabeled data and clustering analyses endeavor to segment data into sub-groups that share similar characteristics. The data in this dissertation is unlabeled because it has not been classified into pre-defined groupings or categories. This contrasts with supervised ML, where models are trained with labeled data to classify new unlabeled data into one of the learned categories.

Consequently, this analysis tested a variety of feature extraction methods, evaluated their performance for unsupervised clustering tasks, and selected the highest performing model using appropriate evaluation criteria. The data in this project contains categorical, numeric, and free text features. The original goal of the analysis was to develop several clustering models with both the categorical and numeric features, then with the free text features, and determine the best performing model based on a common criterion. However, the final models focus only on the free text features due to significant data quality issues with the categorical variables. The rationale behind the decision to pursue a Natural Language Processing (NLP) framework is detailed further in the methodology section.

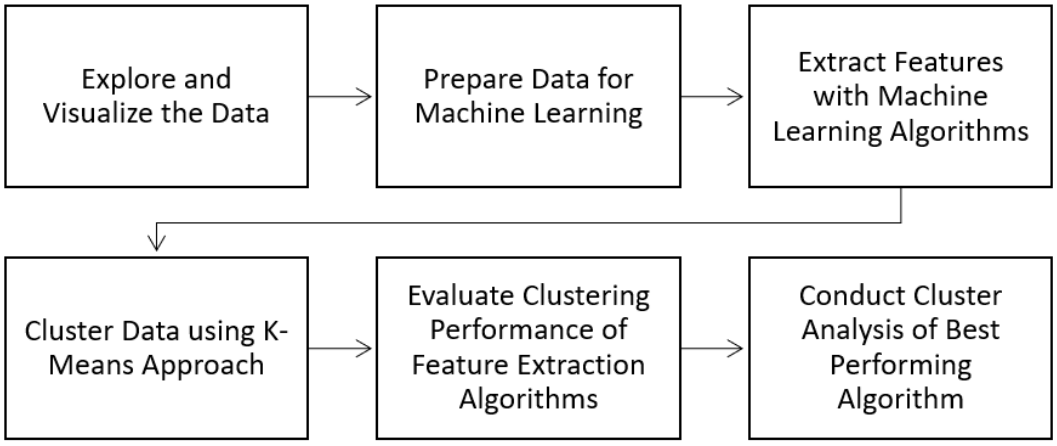


Figure 1: Analysis Process Flow.

2. Technical Background

This section surveys the fundamental mechanics behind the relevant methods in the areas of feature extraction frameworks, clustering models, and neural networks in an NLP context. This is not a comprehensive review of every valid method in these fields but does cover the methodologies considered for this analysis. The methodology section provides greater detail of the rationale behind the choices to include or exclude the methods surveyed here in the final analysis.

2.1 Feature Extraction from Natural Language

2.1.1 Vectorization

Text vectorization, the process of converting free text input data into numeric vectors or arrays, is one of the fundamental approaches to natural language feature extraction (Jurafsky and Martin, 2014). Feature extraction transforms raw data by reducing its dimensions and formatting it so that the data can be processed by ML models. The simplest way to accomplish this is via one-hot encoding, which creates a binary matrix of each row and unique value present in the data set. Each cell is then populated with a "1" if the value in that column is present in the corresponding row of data (referred to as a document), or "0" if the value is not present. This framework is most used for categorical features, but its principles can be adapted to free text data using the bag-of-words method (Doermann, 1998).

Bag-of-words models employ the same matrix structure as one-hot encoding, but instead of populating the array with binary values based on the presence of a given value, the cells represent the frequency with which the value appears in the corresponding document (Doermann, 1998). This approach is more appropriate for free text features because, unlike categorical features, the frequency which a word appears in a block of text is part of the signal as to how meaningful the word is in relation to the text. However, this framework still does not account for the semantic ordering of words nor for the frequency of words in a document relative to the frequency of the same words in the entire text corpus. Additionally, because bag-of-words models generate a matrix column for every unique value in a corpus, these models perform poorly on very large data sets due to the problem of high dimensionality (Aggarwal and Zhai, 2012). The Term Frequency-Inverse Document Frequency (TF-IDF) model improves upon bag-of-words models by accounting for relative frequencies of words in the context of the entire corpus (Christian, Agus, and Suhartono, 2016). The methodology section provides further details on the TF-IDF model and justifies its inclusion in this analysis.

2.1.2 Word Embedding

Just as the TF-IDF model circumvents the relative frequency problem of bag-of-words models, the word embedding framework addresses the issue of high dimensionality (Le and Mikolov, 2006). Word embedding entails transforming raw data into a vectorized format, but instead of a vector the length of the number of total unique values, the vector has a pre-defined length that is much shorter than the total unique value count. Figure 2 visualizes this concept. This is achieved by employing a dimensionality reduction technique, such as principal component analysis (PCA), to map the raw data onto a feature space and create and populate the matrix using the coordinates of each data point. The embedding scores that populate the matrix can be generated by models working only with the data at hand, or larger models pre-trained on massive amounts of free text data. Examples of such pre-trained models include Word2Vec (Mikolov, Corrado, and Chen, 2013), GloVe (Pennington et al., 2014), and fastText (Bojanowski et al., 2016), all employing similar strategies only with varying hyperparameters. This project focuses on the paragraph vector, or Doc2Vec, model, which is detailed further in the methodology section.

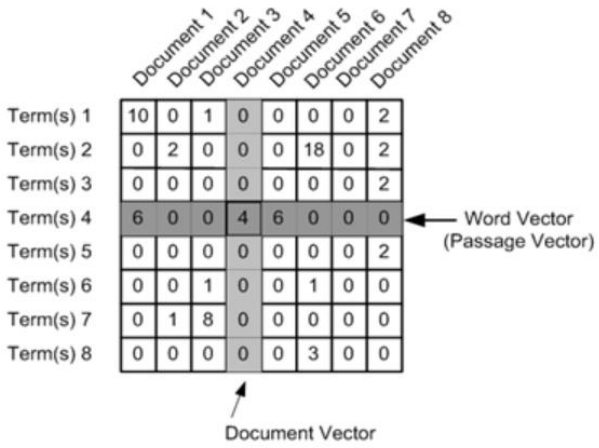


Figure 2: Word Embedding Diagram (Sarwan, 2017).

2.1.3 Deep Neural Networks

Deep neural networks (DNNs) —neural networks consisting of multiple stacked layers— are an alternative learning approach to more conventional ML frameworks. The fundamental structure of a neural network is a connected grid of nodes that combine input data with sets of specified weights that either amplify or nullify the signal gleaned from the input, as shown in figure 3. Each node is assigned a significance weight which is summed and passed through an activation function, determining whether a given node is activated and the extent to which the signal advances through the network (Jurafsky and Martin, 2014).

DNNs are differentiated by their depth, where each layer produces a distinct feature set that is fed into the subsequent layer and recombined with features from other previous layers, until an output value is produced for each data point. This structure is referred to as a feature hierarchy, where the feature representation in each layer of the network increases in complexity as the layers get further from the input layer (Liang et al., 2017). In NLP, the input layer consists of raw text data, the middle layers perform automatic feature extraction via the process just described, and the output values are used for a variety of tasks from classification and prediction to becoming the inputs themselves for other models, such as clustering algorithms (Liang et al., 2017). In this feature extraction process, the linearity of the data flow from layer to layer is what classifies neural networks as feed forward networks.

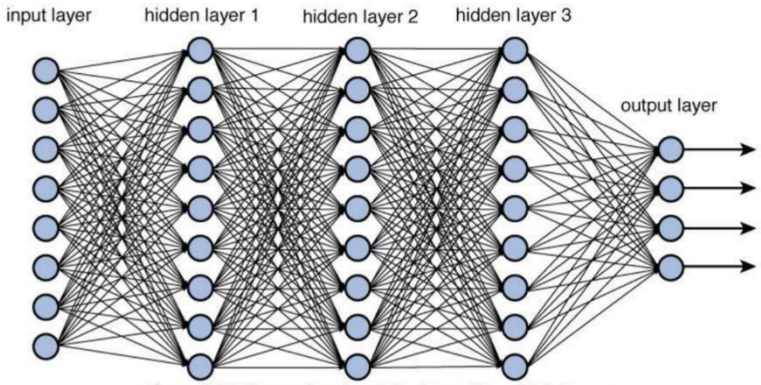


Figure 3: DNN Diagram (Parmar, 2018).

Recurrent neural networks (RNNs) build on this framework by incorporating long-term dependencies into the networks hidden state (Pearlmutter, 1989). This means that instead of each layer receiving one set of inputs generated by the previous layer, the layers receive a sequence of inputs consisting of the inputs generated by the previous layers and the states of those inputs from when they were processed by layers earlier in the model. This memory function is what produces the long-term dependencies where weights assigned to nodes later in the model are functions of how that data was previously processed (Mikolov et al., 2011). This structure allows for parameter sharing- the notion that the looping structure of RNNs allows them to share the weighted parameters each layer applies to the data across different time steps. The memory function and parameter sharing features of RNNs are advantageous because they decrease the computational cost of training by reducing the number of parameters. They also allow the model to capture sequential information in the data, such as the semantic ordering of words in a sentence. Figure 4 demonstrates how the memory function creates the looping structure integral to RNNs.

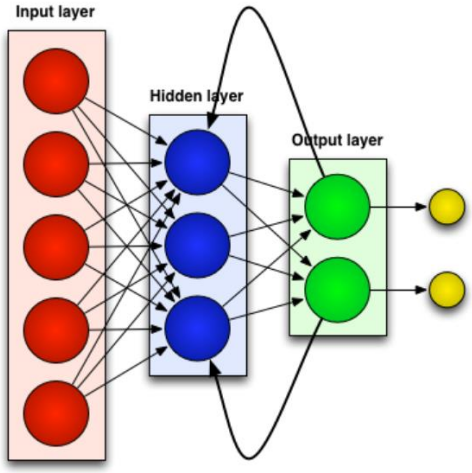


Figure 4: RNN Diagram (Roell, 2017).

Convolutional neural networks (CNNs) have more computational power than RNNs, but conventionally CNNs have only shown superior empirical performance on classification and prediction tasks using visual or imaged-based input data (LeCun et al., 1989; LeCun et al., 1998). The fundamental structure of a CNN consists of stacked convolutional and pooling layers of neurons which receive and transform signals from the previous layer's neurons. Convolutional layers contain filters, or kernels, that scan the feature space of the previous layer and capture only the features within the pre-set size of the filter matrix. The neurons located in these sub-spaces connect the layers within the model and allow layers located closer to the input layer to capture higher-level features, and layers closer to the output layer to capture lower-level features. This framework permits the model to bias filters that maximally learn the signal from localized areas of the feature space (Gu et al., 2018). To increase performance, activation functions can be included between convolutional layers to introduce non-linearity, allowing the model to interpret a higher degree of complexity in the feature space. Pooling layers help reduce the computational load of the model and aid in extracting the features most dominant in the overall signal. Once the signal is passed through the stack of layers, the extracted features may be fed into other ML models or passed through a final classification layer within the CNN. Figure 5 demonstrates how the connections between CNN layers and the signal from input data is transformed into features suitable for ML tasks.

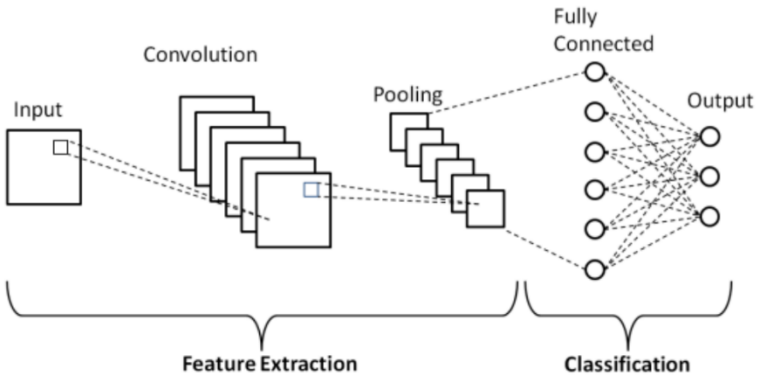


Figure 5: CNN Diagram (Phung and Rhee, 2019).

2.2 Clustering Methods

2.2.1 Centroid-Based Clustering

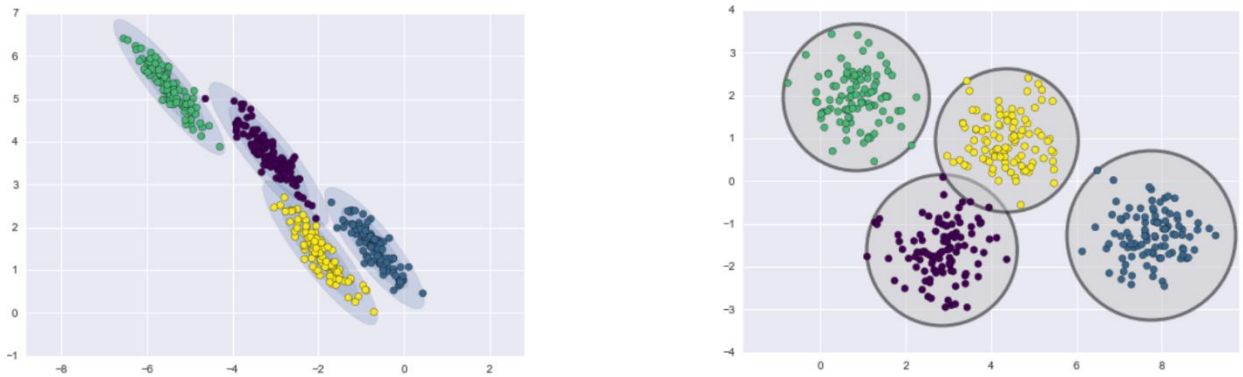
Centroid clustering algorithms organize data into non-hierarchical clusters using distance-based measures. The most common of such algorithms is k-means, due to its computational efficiency and simplicity to implement (Mathur and Kaushik, 2014). K-means algorithms cluster data by randomly initializing ‘k’ number of centroids and assigning each data point to a centroid, so to minimize the mean intra-centroid sum of squares distance. The algorithm then iteratively refines the location of each centroid based on the data point locations from the previous centroid until the clusters fully stabilize. The most common distance measure in k-means clustering is Euclidian distance, the length of a line segment between two points in a Euclidian space. In the context of k-means, the Euclidian distance is the line between a given data point and the cluster centroid, with cluster stabilization occurring when the average Euclidean distance for each cluster is minimized.

There are variations of the k-means framework that employ different centroid initialization methods apart from random assignment, the most relevant of which are detailed in the methodology section. K-means clustering is categorized as a hard clustering approach, meaning that each data point is strictly assigned to cluster. An alternative approach for data sets with a high degree of overlap is fuzzy c-means, which replaces the binary assignment of k-means with a continuous interval that assigns each data point a probability of being assigned to a given cluster (Cannon, David, and Bezdek, 1986). The primary drawbacks of centroid algorithms are that because they are based on mean distance, they are very sensitive to outliers and do not work well

with categorical data. These issues can be mitigated by outlier removal (Zhao, Liang, and Cao, 2013), data normalization practices (Patel and Mehta, 2011) and by assigning numeric category codes to categorical features.

2.2.2 Gaussian Mixture Models

Gaussian mixture models cluster data in a similar manner to k-means algorithms but without the same limitations. For example, fuzzy c-means clusters data points by assigning values based on the probability of that data point belonging to each cluster, gaussian mixture models identity the probability that each data point belongs to a given gaussian, or normal, distribution (Rashid Ahmed Ahmed et al., 2019). This means that gaussian models can account for variance whereas centroid models cannot, better equipping gaussian models to handle data that is irregularly shaped or clustered in a non-spherical way. The reason for this is that centroid-based approaches create clusters by forming a circle around each centroid via a radius defined by the data point furthest from the center. Figures 6 and 7 demonstrate how gaussian models' consideration of variance equips them for clustering non-spherical data sets. Gaussian models have similar disadvantages to centroid-based models and have an additional limitation in that they are only able to perform hard classification, whereas centroid-based models can perform both hard and soft classification.



Figures 6 and 7: Non-Spherical Clustering (GMM) and Spherical Clustering (K-Means) (Maklin, 2019).

2.2.3 Hierarchical Clustering

Hierarchical clustering employs dendrograms, or tree diagrams, to cluster data sequentially. This process can take two forms: agglomerative and divisive. Agglomerative clustering, or bottom-up clustering, is the most common form of hierarchical clustering approaches (Mathur and Kaushik,

2014). Beginning with each data point as its own cluster, the model iteratively creates additional clusters by joining each data point with its nearest neighbor, until a single cluster containing all the data points remains. Divisive clustering, or top-down clustering, is the inverse, starting with one cluster containing the entire data set and iterating down until each data point is its own cluster. The distances between data points are determined via linkage methods. Common linkage methods include:

- Single linkage: cluster by the distance between the nearest single data points of two clusters
- Average linkage: cluster by the shortest distance among all observed cluster pairs
- Ward linkage: cluster by minimizing the sum of squared distances within all clusters
- Complete linkage: cluster by minimizing the longest distance among all observed cluster pairs

Hierarchical clustering is useful because of the visual aids provided by dendrograms, and the flexibility in distance-based clustering linkage methods. However, the framework does not scale well on larger data sets and can be prone to poor accuracy scores due to misclassification, as once the data is split it cannot be recombined (Mathur and Kaushik, 2014).

2.2.4 Density-Based Clustering

Density-based clustering algorithms cluster data around high point density areas on a feature map that represent coherent clusters, often referred to as neighborhoods, separated by the contiguous regions of low point density. The most common of such algorithms is Density-Based Spatial Clustering of Applications with Noise (DBSCAN). The DBSCAN framework is designed to correct for the poor performance of distance-based clustering algorithms on irregularly shaped feature spaces (Ram et al., 2010).

DBSCAN uses two parameters: the radius of a neighborhood around the center data point (ε), and the minimum number of data points allowed to populate a neighborhood (*minpts*). These parameters serve as inputs into the model that categorizes data points into three groups: core points, border points, and outliers. Core points are data points that satisfy the minimum density requirement for a neighborhood, set by the minpts parameter, and which keeps the cluster mass at a level ensuring a given density threshold. Border points are the remaining points in a

neighborhood that contribute to its density calculation but are not defined as core points. Finally, outliers are points that are not assigned to any cluster because they reside in the low-density spaces of the feature space. The DBSCAN framework selects data points at random and computes their assigned neighborhood to determine whether it is a core point. If the data point is designated as a core point, a cluster of other core points and border points will be added around it until the outermost points begin to be defined as outliers. This process is repeated until all data points in the feature space are classified.

Unlike distance-based models, density-based approaches do not assume that the cluster centroids are the center of gaussian spheres, allowing them to better accommodate clusters of different sizes, densities, and shapes. Other advantages of density-based approaches include the ability to detect noise in data via outlier classification and the lack of a requirement to select the number of clusters prior to running the model. Despite these benefits, density-based models do not perform well on high-dimensional data sets nor on data sets with a high variance in cluster density.

3. Literature Review

This section surveys relevant literature in the NLP, text clustering, and public health spaces. Homelessness is often viewed through a public health lens, so it is appropriate to include public health research in this review. Table 1 provides an overview of the parameters used in this literature review. Specifically, these parameters are designed to address the questions of 'what are the most recent and influential advances in NLP deep learning approaches?' and 'how have feature extraction and clustering methodologies been applied in the public health field?'. Only papers published since 2015 are included in the review because of the rapidly evolving nature of NLP and ML literature.

Table 1: Literature Review Search Parameters.

Digital Libraries	UCL Research Library
Search Terms	'NLP Feature Extraction', 'NLP Healthcare', 'Unsupervised Text Clustering', 'Text Analytics Public Health', 'NLP Transfer Learning'
Search Items	Title, Abstract, Keywords
Document Types	Journal Articles, Conference Papers
Timeframe	2015-2021

3.1 Related Works in Natural Language Processing

3.1.1 Configuring CNNs for NLP

The development of the dynamic convolutional neural network (DCNN) established a framework that accurately represented natural language data and successfully extracted features that achieved robust performance on classification and sentiment prediction tasks. DCNNs consist of one dimensional and dynamic k-max pooling layers, defined as "a generalization of the max pooling operator, a non-linear subsampling function that returns the maximum of a set of values" (Kalchbrenner, Grefenstette, and Blunsom, 2014)[2]. Multiple studies have built upon this work and adapted CNNs in the NLP space. In 2015, the DCNN was adapted for relational feature extraction via positional word embeddings within sentences (Nguyen and Grishman, 2015), and more recently in 2019, a CNN was configured to perform normalization and classification of sentiments for unstructured sentences (Arora and Kansal, 2019). Arora and Kansal (2019) explored the polarity of sentiments expressed in social media posts from Twitter and Facebook users. The authors conventionally prepared and normalized raw social media post data for ML, then used a character-level embedding framework to feed features into a CNN, which categorized records as 'positive', 'negative', or 'neutral'.

3.1.2 Transformer Models

Transformer models were developed as an alternative deep learning approach to recurrent and convolutional neural networks configured in an encoder-decoder framework. Transformers do not rely on recurrence or convolutions and are solely based on attention mechanisms (Vaswani et al., 2017). Attention mechanisms allow transformer models to focus only on the most relevant elements on the input sequences, rather than the entirety of the sequence, as is the case with DNNs. This is achieved through a process of self-attention, wherein each input encoder is assigned a score which is then multiplied by the SoftMax score of the encoder. The transformer models' ability to handle variable sized inputs by stacking self-attention layers, as opposed to using neural networks, results in increased performance on generalized learning tasks.

Transformer architecture is advantageous in relation to RNN architecture because leverages parallel processing, as opposed to sequential processing, which allows the layers to interact without

[2] This study is outside of the literature review timeframe, but is foundational in configuring CNNs for NLP tasks, thus is appropriate to include in this review.

passing through many other layers (Merkx and Frank, 2021). Additionally, transformer models perform better on sequential tasks because they can learn long-range dependencies. On tasks involving processing a set of objects, they avoid making assumptions related to the temporal and spatial relationships of the data. However, if the input data contains inherent temporal and spatial relationships, such as free text data, the lack of assumptions will hurt model performance. In such cases, positional encoding must be employed to account for this model shortcoming.

The most influential sentence transformer models are BERT (Bidirectional Encoder Representations from Transformers) and RoBERTa (Robustly Optimized BERT Pre-Training Approach) (Stoyanov et al., 2019). BERT is language representation model developed by Google that "is designed to pretrain deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers" (Devlin et al., 2019). Bidirectional representations allow the model to read the entire corpus of input text data at once. This process is a departure from the directional approach of previous transformer frameworks, which constrained models to encode the input text accounting for direction (i.e., right to left or left to right). BERT achieves bidirectional representations via two methodologies: masked LM and Next Sentence Prediction (NSP). Masked LM is a step in the learning pipeline before BERT is fed word sequence inputs in which 15% of the words in each input sequence as replaced by an unidentified, or masked, token. Multiple layers, including a classification and a SoftMax layer, are then added to the model to use the remaining words to predict the words obscured by the masked tokens. BERT then takes the predictions of the masked words into account, slowing the model convergence time and improving performance on predictive NLP tasks.

Similarly, NSP is a step in the BERT training process in which the model is fed sentence pairs as inputs and is trained to predict whether the second sentence is related to the first. Positional embedding tokens are assigned to each sentence and half of the sentences are paired sequentially while the other half are paired randomly. The combination of training BERT using NSP and masked LM minimizes the combined loss function of both methods. The RoBERTa model, developed by Facebook, is conceptually similar to BERT but improved overall performance by removing the NSP function from training and replacing it with dynamic masking. Dynamic masking is a similar strategy to masked LM but differs in that the masked tokens are not static and change throughout the training epochs (Stoyanov et al., 2019). The development of BERT and

RoBERTa prompted growth in the sentence transformer model space in the form of derivative models tuned to optimize performance on specific NLP tasks, such as topic modeling and free text clustering. As described in the methodology section below, this analysis employs such models designed for free text clustering.

3.1.3 Short Text Clustering

Conducting clustering analyses on data sets with short documents presents a challenge due to the spareness of the text representation, as many words only appear once in each document. The data sets used in this project fit into the short text category, as most of the records are documents containing fewer than fifty words. Various methods have been tested to circumvent this issue, including the use of self-taught CNNs (Xu et al., 2016) and frequent closed word sets (Bai and Jin, 2019). The work of Xu et al. produced a framework for short text clustering $(STC)^2$, a flexible self-taught neural network that can learn non-biased deep text representation and incorporate semantic features of the text corpus. They accomplish this by embedding the raw text features using dimensionality reduction, then feeding the word embeddings into convolutional neural networks to learn the deep representational space. These outputs are then to fit k-means clustering models and the model accuracy results are evaluated against other feature extraction methods, such as Doc2Vec, SkipVec, and recursive neural networks. The results of the analysis reported superior accuracy of the STC^2 framework as compared to other approaches.

The work of Bai and Jin (2019) presented a framework referred to as micro-blog STC, which expresses each document as a frequent word set. This approach created an overlapped text corpus via the following steps: mining out all the frequent word sets, partitioning each word set to create clusters of all documents that are covered by a frequent word set, and divided the word sets into the most distinct clusters by calculating as similarity score. This framework was tested against traditional k-means clustering and performed better based the precision and recall evaluation metrics.

3.2 Related Works in Public Health

In recent years, much work has been done in the public health domain relevant to the scope and goals of this project. Specifically related to homelessness, a 2019 study of people who visited emergency homeless shelters in Ireland from 2012-2016 employed k-means clustering to determine the presence of any sub-groups among the population (Redmond, O'Donoghue-Hynes,

and Waldron, 2019). The results of the study found three distinct clusters representing people experience unabridged short-term homelessness, abridged short-term homelessness, and chronic long-term homelessness. Although the clustering algorithms in this study achieved high performance, the scope of the results was severely limited by feature availability, as the data set only included information of the number of total nights and frequency of stays each person spent at a shelter. This study provided insight into clustering with low quality data sets by enumerating approaches for feature cleaning and selection.

Elbattah and Molloy (2017) employed a k-means clustering approach to patient segmentation in the context of hip fracture care in Irish hospitals. The study used medical record data to group elderly patients by age, length of hospital stays and elapsed time to surgery, then used unsupervised clustering to explore correlations relating to patient demographics and health outcomes. Another example of unsupervised clustering in the healthcare space is the work of Stevens et al., (2019), in which hierarchical agglomerative clustering were applied to identify behavior typologies in children with autism spectrum disorder. The authors employed a Gaussian Mixture Model for feature extraction on a data sample of over 2000 medical records and their analysis produced sixteen distinct sub-groups. The pre-recorded treatment response features for each sub-group were analyzed via regression to gain insights into tailored treatment plans based on patient characteristic. Each of these studies informed this analysis by offering strategies that incorporated text and non-text data, and by providing a comparison of k-means and hierarchical clustering methods.

Exploring the NLP feature extraction space, Arnaud et al. (2020) predicted hospital triage by integrating and extracting features from structured and unstructured data. The data set used in this study contained categorical, numeric, and free text features. The authors extracted features from the numeric and categorical data via a standard multi-layer perceptron (MLP) model and further extracted features from the free text data using a convolutional neural network. The outputs of these two independent processes are then concatenated to form a global feature map, which is then fed into another MLP model to perform the classification task. Key insights from this study were the pre-processing pipeline created for free text data and the construction of a CNN for the purpose of free text feature extraction. Chen et al. (2020) also proposed an automated NLP feature extraction framework to decode features sets of medications and reasons for their prescription to

patients. The authors demonstrated the efficacy of their approach using unstructured data collected from two epidemiological studies of myalgic encephalomyelitis/chronic fatigue syndrome. Implementing the automated NLP pipeline reduced the manual processing time of mapping medication names to their uses for 84% and reduced the time taken to process unstructured text data by 91%. This dissertation aims to create a similar automated NLP pipeline to reduce the manual processing load for Centrepoint and allow them to match causes of homelessness more efficiently and efficiently to sub-groups of their beneficiaries.

4. Methodology

4.1 Overview of Datasets

This project employed two data sets, both containing interactions from the Centrepoint crisis helpline. One comprises conversations directly between a Centrepoint representative and a young person in crisis (referred to as the helpline data set) and the other comprises conversations between a representative and an advocate speaking on behalf of a young person (referred to as the enquiry data set). The data collected encompasses records from January 2017 -- May 2021. Although there is significant feature overlap among the data sets, there are key discrepancies both in terms of the number of total features and in how the data is recorded in certain features with the same name. For these reasons, the data sets could be combined, and the models were run and evaluated against each data set separately. Table 2 provides a high-level overview of each data set.

Table 2: Data Set Summary.

Data Set	Record Count	Total Feature Count	Categorical Feature Count	Numeric Feature Count	Free Text Feature Count
Helpline	16,309	47	44	1	2
Enquiry	6,932	62	58	3	1

4.2 Data Quality Issues

As table 2 shows, most features are categorical. However, both data sets are treated as mixed data sets because of the business importance of the non-categorical features and the high number of null values in many of the categorical features. In terms of the non-categorical features, the

numeric feature (in this case, age) and the free text feature (the enquiry description) contain vital information about demographic profiles and housing situations and therefore should be considered in any clustering analysis. From a data quality perspective, most categorical features in both data sets are not suitable for ML because of untenably high rates of null values. After replacing non-pertinent values, such as 'N/A' or 'unknown', with nulls and filtering out features with null value rates greater than 80% for non-text fields, only 5 of the 47 features in the helpline data and 4 of the 62 features in the enquiry remained usable. Table 3 is a sample of the helpline data set after the filtering process and shows the severe limitations in scope the null values problem causes.

Table 3: Usable Categorical Features in the Helpline Data Set.

	Housing Situation	Gender	Region	Main cause of homelessness	Age	Status
1	Living with family	Male	None	Family breakdown	21.0	At-risk
2	Living with family	Male	None	Other (please specify)	27.0	At-risk
5	Private sector tenancy	Female	None	Eviction	25.0	At-risk
7	Living with family	Female	South East England	Family breakdown	19.0	Experiencing Homelessness
8	Sofa-surfing	Female	South East England	Eviction	22.0	Experiencing Homelessness

The free text features in each data set also had a significant number of null values, but each contained over 6000 remaining records suitable for ML. Although it is limiting that this analysis could not include many features, the issue is understandable given the context the data is collected in. The interactions the helpline capture involve people who are vulnerable and amid housing crises and the priority is to focus on resolving the situation as quickly as possible. Consequently, many conversations are shorter in length and only capture basic information before the interaction is ended or moves in a more solution-orientated direction. It is for these reasons that the aim of the analysis shifted from mixed data set feature extraction to strictly free text feature extraction model evaluation.

4.3 Exploratory Data Analysis

Descriptive data exploration through visualization is an important step in better understanding the context and content of the data, prior to preparing it for ML models. This section provides insights into the demographics and housing situations of Centrepoint's beneficiaries and identifies patterns

evident in both text and non-text features. Due to poor data quality in the categorical features in the enquiry data set, the non-text visualizations only represent records from the helpline data set. The figures representing free text features denote which data set is being depicted.

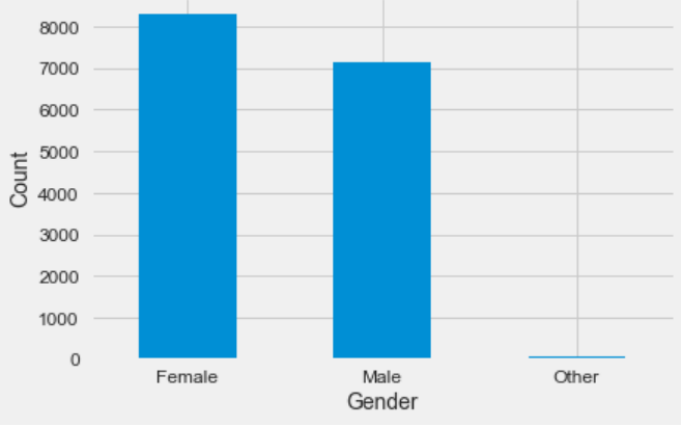


Figure 8: Counts by Gender.

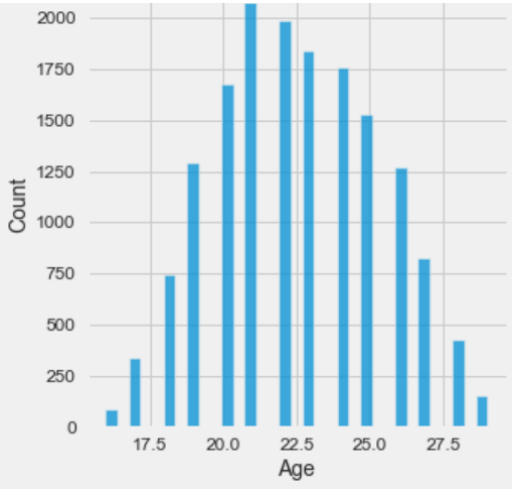


Figure 9: Age Distribution.

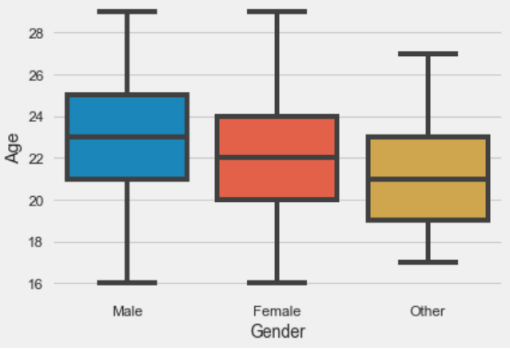


Figure 10: Age Distribution by Gender.

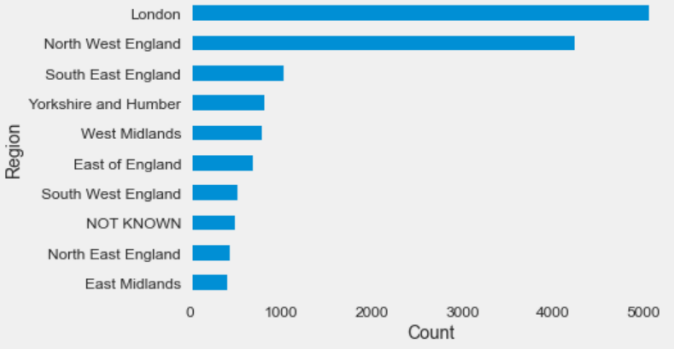


Figure 11: Counts by Region.

Figures 8, 9, 10, and 11 summarize the demographic makeup of Centrepoint's beneficiaries. The data skews slightly female, and those who identify as female or other skew younger than those identifying as male. Overall, the age distribution follows a normal curve, given the constraint that the range is limited to ages 16-24 (though some aged above 24 are included in the data). The regional distribution is also proportionally distributed in that the major population centers of the United Kingdom- London and Manchester (included in the Northwest England region)- comprise the majority of records in the data. Similarly, the housing situation data is skewed toward one or

two leading categories per feature. As figures 12 and 13 demonstrate, the most common situation is a young person currently experiencing homelessness, who contacted the helpline due to a family breakdown.

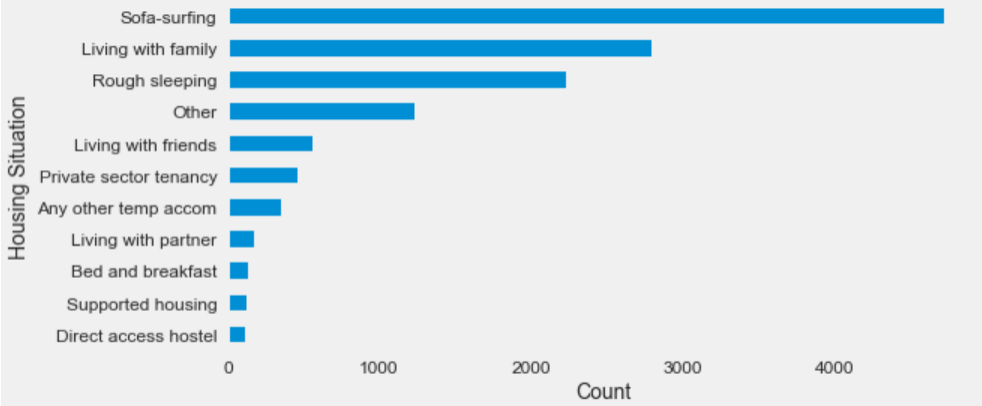


Figure 12: Top 10 Housing Situations.

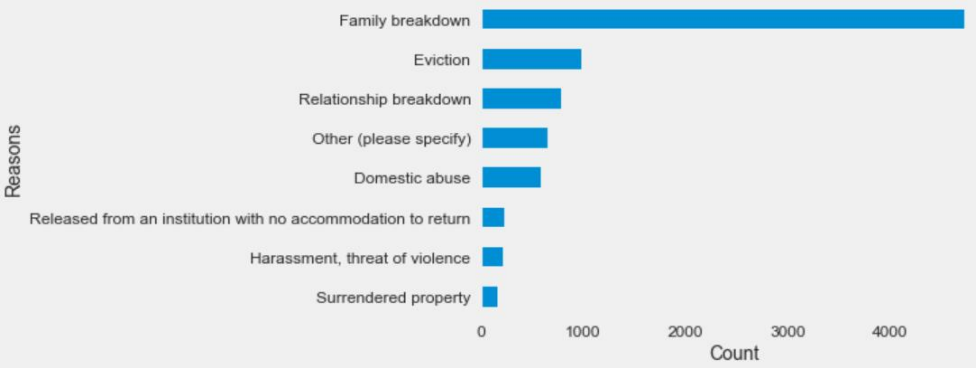


Figure 13: Top 10 Reasons for Contacting the Helpline.

Figures 14 and 15 provide an overview of the free text features and hint at the subtle difference between the helpline and enquiry data sets. Both sets are highly skewed to entries under fifty words, however, the enquiry data has a much longer tail towards lengthier entries. The word clouds in figures 16 and 17 also demonstrate variation as the most common words in the helpline data, such as ‘sofa’, ‘surfing’, ‘kicked’, and ‘sleeping’ are more related to housing situations directly, while the most common words in the enquiry data, such as ‘Centrepoint’, ‘application’ and ‘advised’, are more administrative in nature.

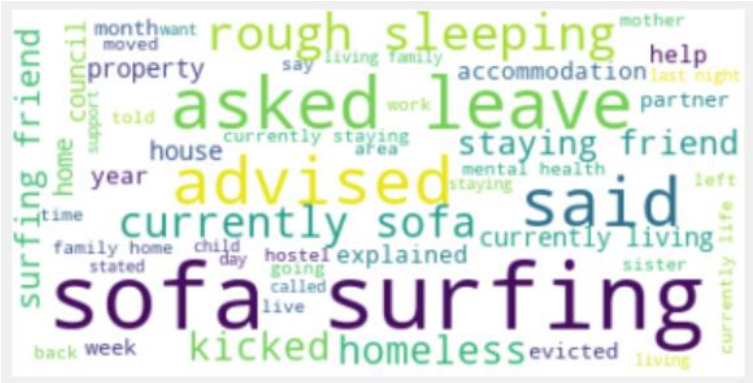


Figure 14: Helpline Word Cloud.

Figure 15: Enquiry Word Cloud.

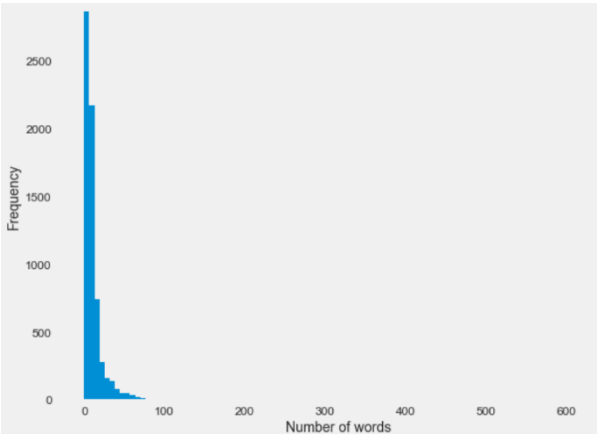


Figure 16: Helpline Average Document Length.

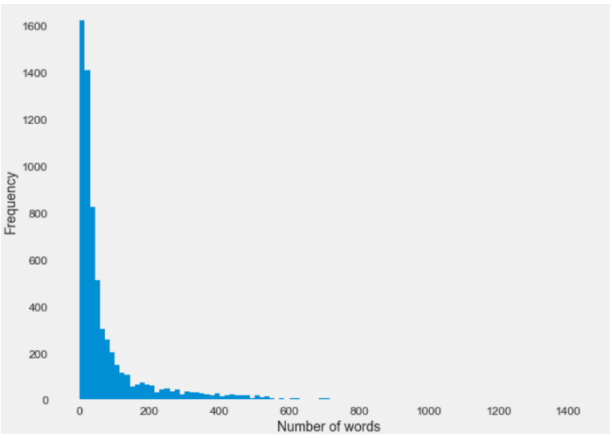


Figure 17: Enquiry Average Document Length.

An analysis of the most common bigrams, shown in figures 18 and 19, corroborate the respective patterns found in the word clouds. Given that the enquiry data consists of conversations with advocates rather than young people, the longer and more secretarial nature of those conversations as compared to the helpline data is consistent with the data collection context.

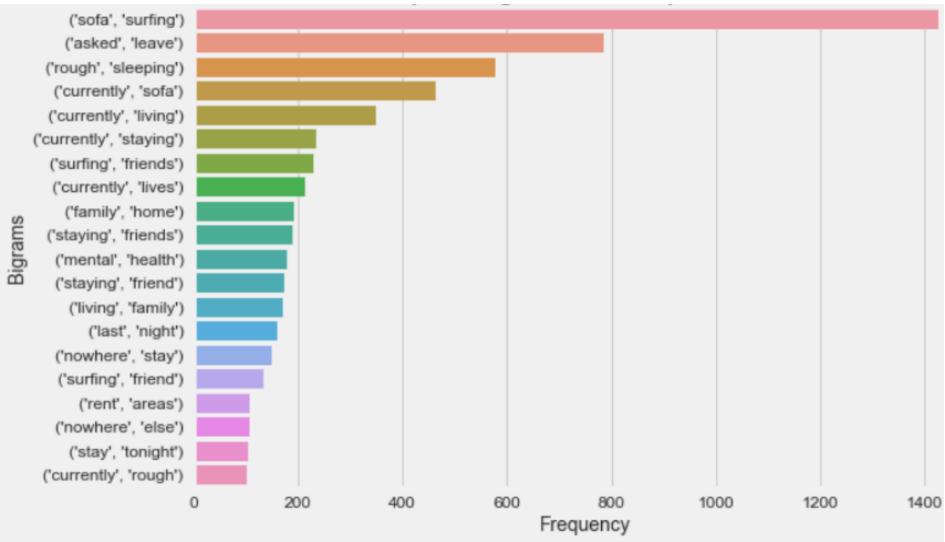


Figure 18: Helpline Most Common Bigrams.

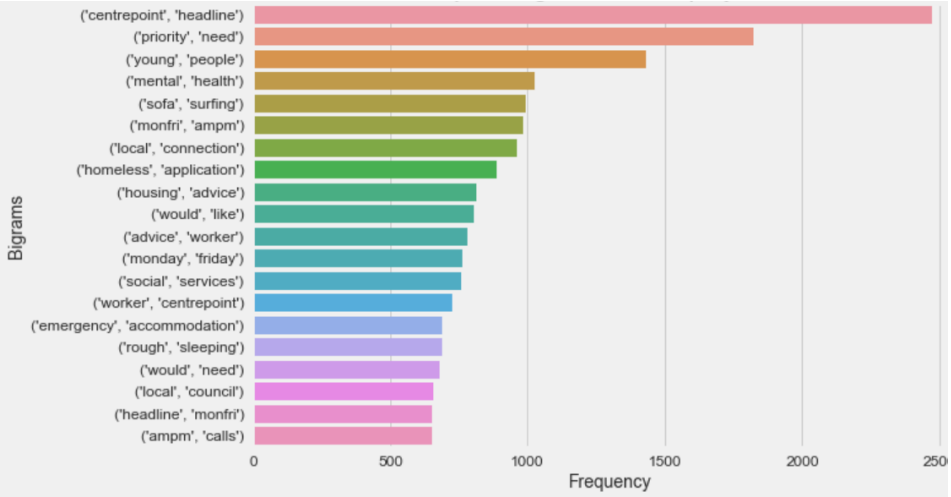


Figure 19: Enquiry Most Common Bigrams.

4.4 Proposed Methodology

The nature of the organizational problem and limitations of the data sets dictated that this analysis employ various natural language feature extraction techniques to fit free text data to unsupervised

clustering models. The result of the analysis is a defined highest performing model, an analysis of the clusters that model produces, and insights into the implications the findings have on the Centrepoint organization. To achieve this result, the three data sets used in this analysis- helpline, enquiry, and a combined set- were defined. The helpline and enquiry data sets both collect interactions from the Centrepoint helpline, however, the helpline data collects conversations directly with young people and the enquiry data collects conversations from advocates on behalf of young people. Although the two data sets contain enough discrepancies that they cannot be wholly combined, it is possible to combine the free text features into one data set, as the content and data collection methods are identical. This fact was verified by the Centrepoint team. The enquiry data set contains one free text feature, while the helpline data set contains two free text features which were combined into a single document as they both describe the housing situation of the young person. The combined data set consists of all records from both data sets.

After the data was defined, it was pre-processed for feature extraction. This process involved the standard steps of spell checking, removing stop words and symbols, tokenization, and lemmatization. The three data sets underwent these processes to produce a final free text feature, referred to as 'final text', for each data set. These features then served as inputs to feature extraction algorithms, which in turn were fit to a clustering model. This analysis tested four feature extraction algorithms: Term Frequency - Inverse Document Frequency (TF-IDF) and Paragraph Vector (Doc2Vec), which employ vectorization, and Distill Roberta and MP-Net, which employ sentence embedding and leverage Transfer Learning. These models were selected for this analysis due to their superior empirical performance on NLP clustering tasks. The studies that demonstrate this performance are enumerated in the feature extraction algorithm selection portion of the methodology section.

The features extracted from each data set by each algorithm were fit to an unsupervised k-means clustering model. K-means was selected as the singular clustering method because the primary aim of this project was to test feature extraction methods. K-means is widely selected as the standard for clustering analyses most similar to this work because of its simplicity to implement, computational efficiency, and consistently strong empirical performance across relevant studies, as the literature review section expounds upon.

After the k-means models were run on the features of each data set, a set of evaluation metrics were used to determine the best performing feature extraction method and the optimal number of clusters for each k-means iteration. A combination of the elbow method and silhouette score established the optimal number of clusters, while a combination of the silhouette score, Davies Bouldin Index, and Calinski Harabasz Index dictated the best performing model. For the optimal number of clusters, if the elbow method and silhouette methods identified different optimal cluster counts for a given model, both were examined and the one that performed the best among the evaluation metrics was selected. Finally, a descriptive analysis of the clusters the best performing model produced was conducted and the limitations of the analysis were examined. The remainder of this section expands upon each part of the methodology in greater detail.

4.5 Data Pre-Processing

The data preparation steps described in the following section are standard practice for pre-processing free text data across supervised and unsupervised NLP tasks. These procedures are designed to remove as much noise as possible from the data so only relevant information is fed into ML models.

4.5.1 Removing Stop Words, Symbols, and Undesired Characters

The first step in the data cleaning process was removing stop words, which refer to the most frequently used words in natural language data. These commonly include short function words with little or ambiguous meaning, such as ‘the’, ‘which’, or ‘at’, which are applied to express relationships among other words in a sentence. Despite this widely accepted definition, there is not canonical set list of stop words in NLP. For analyses employing the Python coding language, the most widely used stop words list for English text come from the NLTK library (Bird et al., 2009), which is the stop word list this project used. In addition to the to the NLTK list, the words ‘young person’, ‘yp’ (an abbreviation for young person in this data), and ‘enquiry’/’inquiry’ (both spellings appear in the data) were added as stop words for this analysis. These words were defined as stop words because they fit the definition as words that do not contribute meaning to a sentence and were present in a very high percentage of the total records. In most contexts, ‘young person’ would be a meaningful term because it describes the age of a person, but in this context all subjects are, by definition, young people, nullifying the descriptive meaning from the term. After removing

stop words, all letters were converted to lowercase and any numeric characters, punctuation marks, and other miscellaneous symbols were deleted.

4.5.2 Spell Checking

Spell checking is another crucial component to NLP and was essential for this project, given the process by which both helpline and enquiry data was collected. Because Centrepoint representatives take hand typed notes when fielding calls, the data is prone to misspellings and other grammatical errors. Like stop words, there is no universal dictionary in NLP that English words are spell checked against. However, Python libraries, 'Spellchecker' (Tiedemann and Lison, 2016) and 'Autocorrect' (McCallum and Sondej, 2021) contain commonly used vocabularies which were relevant for this project. Although there is not a discernible difference in the English language content and functionality of each library, Autocorrect was selected because it performed much faster than Spellchecker. Spell checking did not detect every syntax error in the data, especially those stemming from colloquial usage. However, it corrected enough mistakes to remove significant noise from the data set.

4.5.3. Tokenization

Once the data is cleaned and spell checked, each word must be normalized via tokenization- the process by which a larger bloc of free text data is broken down into singular units, called tokens (Jurafsky and Martin, 2014). This can refer to dividing words into either individual letters or sub-words. In the context of this analysis, however, tokenization refers to word tokenization, in which sentences are broken up into individual word tokens. Each word is defined by the space before and after a set of letters. For example, 'young person' would be classified as two tokens, whereas 'young-person' would be classified as a single token. Tokenization is a necessary step in the feature extraction pipeline for vectorization algorithms, as well more advanced transformer algorithms, thus it is necessary for each extraction technique this analysis employs.

In feature extraction algorithms, each individual token represents a value, and the unique values within the set of tokens creates the vocabulary that the algorithm uses to process the free text data. In vectorization approaches, each token in the vocabulary is treated as a unique feature. A matrix totaling the number of unique tokens per document and the frequently each token appears in each document serve as inputs to the model. In Deep Learning approaches, the vocabulary is used to create tokenized input sentences, which in turn serve as inputs to the model. The primary

disadvantage of word tokenization is that is does not handle out-of-vocabulary (OOV) words (words not represented model's vocabulary) well. This presents an issue for supervised learning methods, such as classification or prediction tasks, that involve test data which contains OOV words. However, for unsupervised tasks that do not include test data, OOV words are not a concern. For this reason, OOV words were not of concern for the data set analyzed in this project.

4.5.4 Stemming and Lemmatization

As a final step in the pre-processing process, a lemmatization technique was used to reduce words to their root by removing conjugative, inflective, and derivative endings. Lemmatization was used instead of stemming because of its ability to return more accurate lemmas, or linguistic root words, compared to stemming. In a linguistic sense, stemming reduces words to pseudo-stems, meaning that the conjugated or inflected word ending is removed, leaving the root word in place. This method results in the preservation of the linguistically correct root word most of the time, but does leave room for error, as it treats derivational and inflectional variance equally (Jurafsky and Martin, 2014). Conversely, lemmatization only treats inflectional variance, which returns more accurate linguistic root words, referred to as lemmas. For this reason, lemmatization is considered the precise processing method. However, this precision only results in relatively modest empirical gains.

The most common stemming technique is Porter stemming, derived from Porter's algorithm (Porter, 1980). Snowball and Lancaster stemmers are also widely used variations of stemming algorithms. The primary difference between these three methods is the level of aggression in determining how much of the root word should remain. Porter and Snowball stemmers are less aggressive than Lancaster stemming, which often returns very short root words that can be difficult to interpret on their own. There are also several options for lemmatization available in Python, including WordNet from the NLTK library, TextBlob (Loria, 2018), and SpaCy (Honnibal and Montani, 2017). For this project, the WordNet algorithm from NTLK was selected because it is the most used technique in related free text analyses (Manning et al., 2009). Table 4 exemplifies how the pre-processing pipeline transformed the helpline data set (recall, helpline data refers to data gathered from calls directly with young people), with the original text on the left and the final transformed text on the right. The enquiry data (gathered from calls with advocates) and combined data sets were transformed using the same pipeline.

Table 4: Pre-Processed Free Text Features.

	index	spellcheck	text	tokens	larger_tokens	clean_tokens	stem_words	lemma_words	final_text
0	1	young person has been staying with relatives i...	has been staying with relatives in europe for...	[, has, been, staying, with, relatives, in, eu...	[been, staying, with, relatives, europe, years...	[staying, relatives, europe, years, family, br...	[stay, rel, europ, year, famili, breakdown, ta...	[staying, relative, europe, year, family, brea...	staying relative europe year family breakdown ...
1	2	He had no job and went to Malaysia to settle	he had no job and went to malaysia to settle	[he, had, no, job, and, went, to, malaysia, to...	[went, malaysia, settle]	[went, malaysia, settle]	[went, malaysia, settl]	[went, malaysia, settle]	went malaysia settle
2	13	Was in a refuse but had to leave due to someon...	was in a refuse but had to leave due to someon...	[was, in, a, refuse, but, had, to, leave, due,...	[refuse, leave, someone, turning, accommodation]	[refuse, leave, someone, turning, accommodation]	[refus, leav, someon, turn, accommod]	[refuse, leave, someone, turning, accommodation]	refuse leave someone turning accommodation
3	17	young person is currently sleeping in their co...	is currently sleeping in their cousins car af...	[, is, currently, sleeping, in, their, cousins...	[currently, sleeping, their, cousins, after, b...	[currently, sleeping, cousins, evicted, stonep...	[current, sleep, cousin, evict, stonepillow, h...	[currently, sleeping, cousin, evicted, stonepi...	currently sleeping cousin evicted stonepillow ...
4	18	Living with ex - partner's mum, so still expos...	living with ex partners mum so still exposed ...	[living, with, ex, partners, mum, so, still, e...	[living, with, partners, still, exposed, behav...	[living, partners, still, exposed, behaviour, ...	[live, partner, still, expos, behaviour, want,...	[living, partner, still, exposed, behaviour, w...	living partner still exposed behaviour want soon

4.6 Free Text Feature Extraction Algorithms Selection

After the data was fully pre-processed, it was fitted to feature extraction algorithms in preparation for the final clustering. The performance of each of the algorithms detailed in this section was the primary mechanism tested in this analysis. The remainder of this section explains the mechanics behind each algorithm and justifies their inclusion in this analysis as well as any changes to the default model parameters.

4.6.1 Term Frequency - Inverse Document Frequency (TF-IDF)

The TF-IDF algorithm (Salton et al., 1988) transforms free text inputs into meaningful numeric outputs that can be fit to ML algorithms. The difference between TF-IDF and count vectorization is that count vectorizers account for the absolute frequency of words in each document, while TF-IDF models account for the relative frequency of words in a document as related to the vocabulary of the entire corpus of text. This technique provides a weight which measures the importance of the word to each token in the matrix. The weights are derived from the inverse document frequency concept, which is demonstrated by the equation in figure 20.

$$\mathbf{tf}(t,d) = \frac{f_d(t)}{\max_{w \in d} f_d(w)}$$

$$\mathbf{idf}(t,D) = \ln\left(\frac{|D|}{|\{d \in D : t \in d\}|}\right)$$

$$\mathbf{tfidf}(t,d,D) = \mathbf{tf}(t,d) \cdot \mathbf{idf}(t,D)$$

$$\mathbf{tfidf}'(t,d,D) = \frac{\mathbf{idf}(t,D)}{|D|} + \mathbf{tfidf}(t,d,D)$$

$f_d(t) :=$ frequency of term t in document d

$D :=$ corpus of documents

Figure 20: TF-IDF Equation (Vogler, 2014).

Words with higher values of TF-IDF weights are relatively more important, whereas words with lower weight values are relativity less important. This conforms with the logic that in a corpus of text, words which appear very often and very seldomly will not be of much use in detecting patterns. TF-IDF was included in this analysis because this concept is especially relevant in the context of clustering. Highly common words will increase the overlap between clusters, and highly infrequent words will make it difficult for clear, reasonable sized clusters to coalesce. In this analysis, TF-IDF is implemented using the TF-IDF Vectorizer function in the Scikit-learn text feature extraction library (Pedregosa et al., 2011).

For hyperparameter tuning, the 'min' and 'max' df settings were changed from the default values of 0 and 1, to 0.1 and 0.8, respectively. The max_df parameter instructs the model to discard words with a weight above the set value and the min_df parameter does the same for words with a weight below the set value. The settings of 0.1 and 0.8 mean that terms above the 80th percentile in usage and below the 10th percentile in usage were ignored in the analysis. The ngram_range parameter was also changed from the default of (1,1) to a value of (1,3). This parameter sets the lower and upper bound for the size of ngrams that the model can extract. The default value only extracts unigrams, while the value of (1,3) extracts unigrams, bigrams, and trigrams. Both parameters are set as they are because after testing the algorithm with the default parameters and the updated set the results, the updated parameters resulted in better performance, as judged by the pre-defined

selection criteria. The primary disadvantages to TF-IDF are that the framework does not account for the order of the words in the document and does not provide linguistic information about the words, such as how similar they are to other words in document or their meanings.

4.6.2 Paragraph Vector (Doc2Vec)

In a 2006 paper, Le and Mikolov introduced the paragraph vector framework (Doc2Vec), building on the existing Word2Vec framework. Word2Vec was developed as an answer to the limitation of traditional bag-of-words models, such as the TF-IDF vectorizer. Such models cannot account for the linguistic meaning of words and associations between words in a text corpus. Doc2Vec is an unsupervised algorithm that learns fixed-length feature representations from variable-length pieces of texts, such as sentences, paragraphs, and documents (Le and Mikolov, 2006). The advantages Doc2Vec compared to Word2Vec are why it was included in this analysis.

There are two variations of the Doc2Vec algorithm: distributed memory (PV-DM) and distributed bag-of-words (PV-DBOW). The PV-DM approach learns to predict where words should appear in a document based on the context of the surrounding words, whereas the PV-DBOW approach uses the paragraph vector to classify all the words in a document, rather than trying to predict related words. This Doc2Vec parameter was set to the PV-DM to take advantage of its predictive power. The Doc2Vec model was implemented using the Gensim Python library (Rehurek and Sojka, 2011) and kept the hyperparameters, other than the PV-DM setting, as the default values.

4.6.3 Paraphrase Distill RoBERTa and MP-Net

Paraphrase Distill RoBERTa and MP-Net are both variations of BERT sentence transformer models, referred to as Siamese BERT (SBERT) networks. SBERT (Gurevych and Reimers, 2019) was developed to address the extremely high computational power that BERT (Devlin et al., 2018) and RoBERTa (Liu et al., 2019) models require for sentence pair regression tasks. SBERT models leverage Siamese and triplet network structures to produce sentence embeddings that can be compared using a cosine similarity metric. This advancement allowed SBERT models to compete the same tasks as BERT in a minuscule fraction of the time while maintaining comparable accuracy levels. SBERT sentence transformers can be used for a variety of NLP tasks, which has resulted in numerous offshoot algorithms tailored to perform in specific areas. SBERT developers complied evaluations of many of these versions and ranked them according to performance on generalized NLP tasks, speed, specific clustering tasks with NLP, and various other more specific tasks. This

research was conducted with sentence inputs of fewer than 128 words, which is descriptive of the average sentence length of the data used in this analysis. Distill RoBERTa and MP-Net were selected as the tested models because MP-Net performed the best on clustering tasks, while Distill RoBERTa achieved the best overall score (Reimers, 2021). Both models were implemented using the sentence transformer Python library. As these models take advantage of transfer learning, hyperparameter tuning was not necessary.

4.7 Unsupervised K-Means Clustering

4.7.1 Mini-Batch and Accelerated K-Means

The full k-means algorithm produces the most empirically robust results of all k-means iterations, but it also is the most computationally expensive (Aggarwal and Zhai, 2012). For this reason, alternative k-means methods, such as mini-batch and accelerated k-means, can be used on larger data sets to reduce the computational burden. Accelerated k-means (Sculley, 2010) disregards less pertinent distance calculations via the use of the triangle inequality theorem. Triangle inequality asserts that given points A, B, and C in a triangle, the distance AC is always less than the combined distance of AB and BC. Consequently, the accelerated k-means algorithm only uses the shorter AC distances in its calculations. This process results in a faster and more efficient algorithm, but also produces less accurate results.

Mini-batch k-means (He, Chang, and Chen, 2010) reduces the computation load by storing random batches of data of a fixed size in memory and running the model on those batches iteratively until clusters converge. This is theoretically similar to stochastic gradient descent (SGD), but is preferred for clustering applications because the batches of data tend to have less noise than the individual data points in SGD. However, mini-batch k-means produces less accurate results than full k-means, where the entire data set is stored in memory. While the speed-accuracy trade-off both mini-batch and accelerated k-means present may be worthwhile for analyses with very large data sets, the limited number of records and relatively short average document size in this project's data did not warrant the use of abbreviated k-means approaches.

4.7.2 Hyperparameter Tuning

For the final k-means model run on each feature extraction, the tuning of the number of clusters and initialization hyperparameters were experimented with to achieve optimal performance. For

other hyperparameters, such as tolerance, number of initializations, and maximum iterations, the default values in the Python k-means package were maintained, as the data sets did not have any abnormal features that needed accommodation and are not considered very large. To determine the number of clusters, k, the model ran iteratively for a k of 2 to a k of 20, then selected the k for the final model based on elbow method and silhouette score evaluation. The list of cluster counts is terminated at 20 because it is unlikely that the relatively low number of records in each data set would produce a high number of distinct clusters. For the initialization parameter, the default value is 'random', which selects randomly located centroids for each iteration of the model. For this analysis, the initialization parameter was set as 'k-means++', ensuring that the chosen centroids for each model run were a certain distance away from each other. Empirical evidence suggests that this initialization method produces more coherent clusters than randomly initializing the centroids does (Arthur and Vassilvitskii, 2006).

4.8 Evaluation Metrics

Clustering models can be evaluated using either external or internal indices. External indices evaluate models by leveraging information grounded in truth, most commonly in the form of pre-labeled data. Internal indices rely on information intrinsic to a model, such as the position of data points and the distance between them. As this analysis does not contain pre-labeled data, it is most appropriate to evaluate the model using internal indices. The three metrics outlined below are commonly used in unsupervised clustering analyses. The model that performed the best against the following metrics was selected for the final clustering analysis.

4.8.1 Elbow Method

The elbow method is the common heuristic in determining the optimal number of clusters for k-means approaches. The metric takes the form of a graph in which inertia, a measure of internal coherence of the clusters, is mapped for a predetermined set of cluster counts, k. In this context, inertia relates to the within-cluster sum of squares criteria, represented by the equation below. As figure 21 demonstrates, the kink in the graph represents an estimate of the optimal k for that model. The elbow method is simple and useful but has the same drawbacks as does k-means clustering in general: it does not perform well with irregularly shaped data, and it is not a normalized metric.

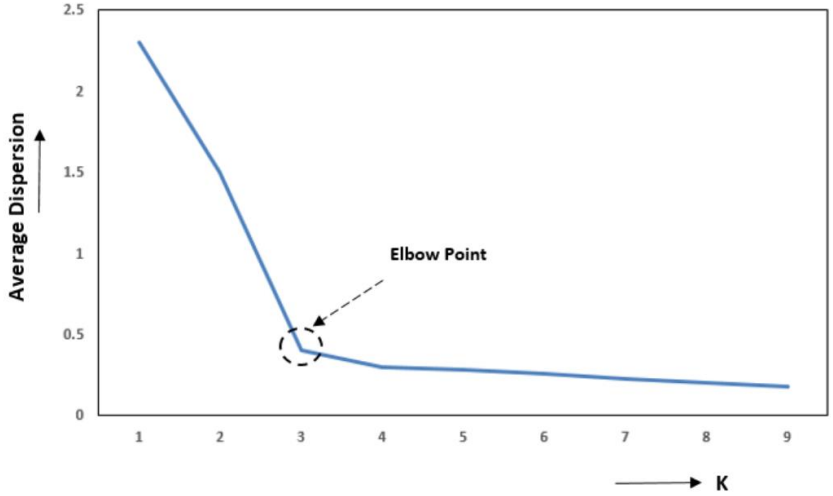


Figure 21: Elbow Method Example for K-Means (Dangeti, 2017).

4.8.2 Silhouette Score

The silhouette coefficient is a metric bound between -1 and 1, where a greater absolute value indicates dense and distinct clusters and values closer to zero indicate significant overlap between clusters. The consistency of the scale is a primary advantage to using silhouette scores; however, the metric can be biased towards convex clusters over other cluster shapes. In practice, a silhouette score is assigned to each data in a cluster and the mean value of all individual scores is the coefficient for the cluster. Mathematically, the metric is calculated by relating the mean distance between a data point and all other points in the same cluster, and the mean distance between a data point and all other points in the next closest cluster.

4.8.3 Davies Bouldin and Calinski Harabasz Indices

The Davies Bouldin (DB) and Calinski Harabasz (CH) indices are often used in clustering analyses because of their mathematical simplicity and low computational cost relative to silhouette scores. Like the silhouette score, the primary disadvantage of each of these metrics is that, like the silhouette score, they can be biased towards convex clusters over other cluster shapes. The DB index is defined as the average similarity between each cluster and the next most similar cluster in the data set, with a score of zero being the lowest possible value and values closer to zero indicating more distinct and coherent clusters. The CH index is defined as the ratio of the between-clusters

dispersion mean and the within-cluster dispersion, with higher scores indicating more distinct and coherent clusters and no maximum score value.

5. Results

5.1 Model Selection

Based on the evaluation metrics, the feature extraction algorithms performed better on the helpline data set than on the enquiry or combined data sets. Overall, a k-means clustering model fitted with text features extracted by the TF-IDF and run on the helpline data set produced the most defined clusters. This model produced ten clusters and had the highest silhouette score and lowest DB index score by a significant margin. The only model to outperform the TF-IDF on any metric was the Doc2Vec algorithm on the CH index, where it achieved a higher score on both the helpline and combined data sets. Figures 22 and 23 demonstrate why ten is the optimal number of clusters for best performing model. The elbow method shows a kink in the graph where k=10 and the improvement in silhouette score begins to level off around the same number k. Numbers just below and above 10 were also tested, but neither produced an improvement in model performance.

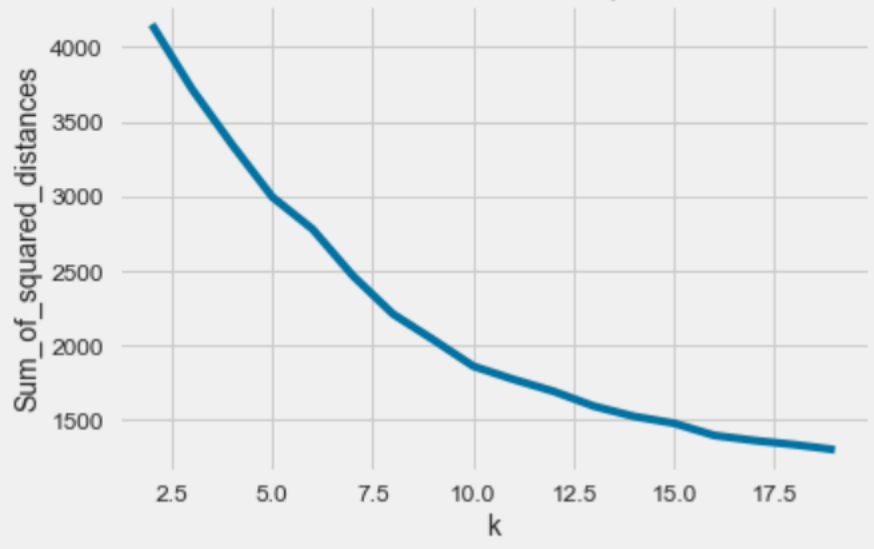


Figure 22: Elbow Method Chart for TF-IDF Algorithm on Helpline Data.

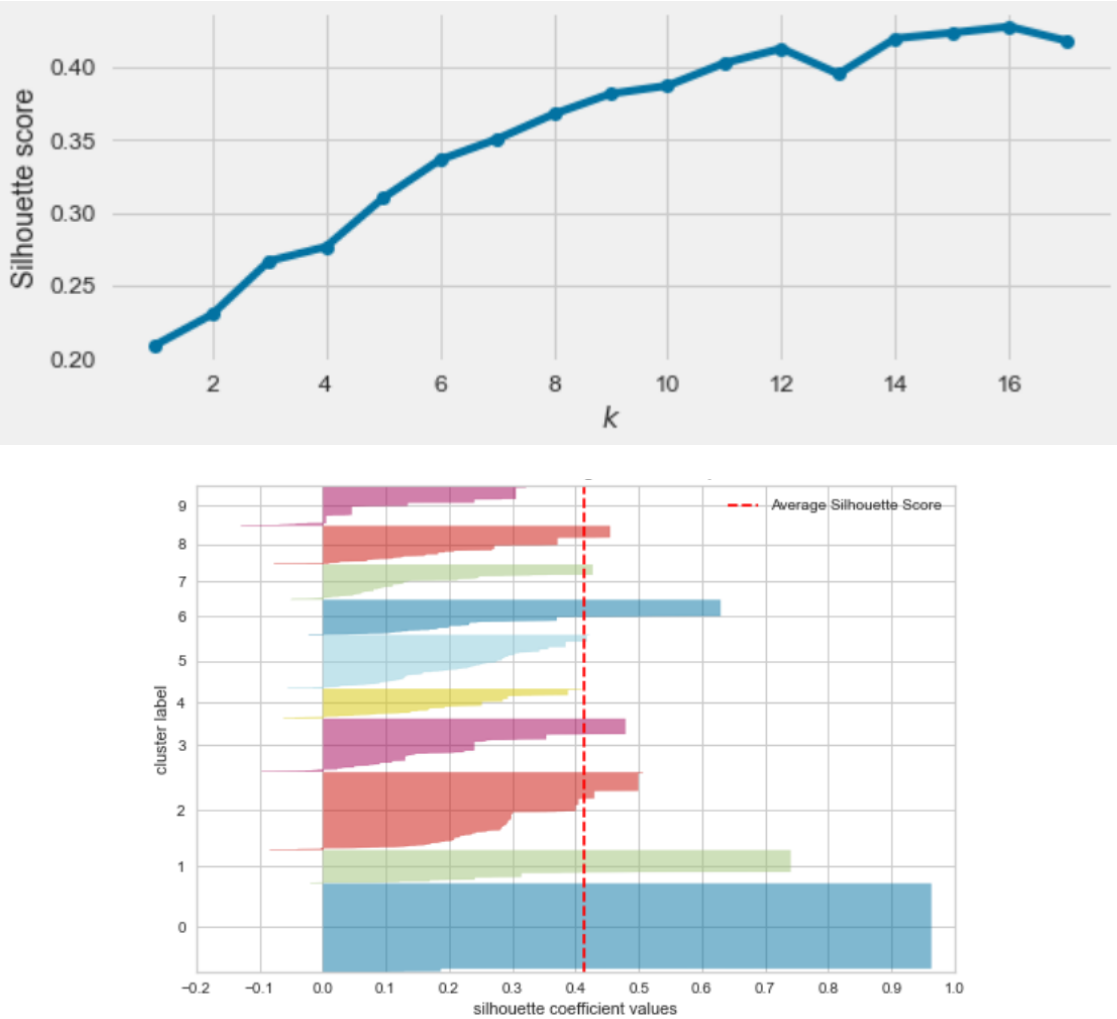


Figure 23: Silhouette Score Charts for TF-IDF Algorithm on Helpline Data.

Table 5 displays the results of the experiments, with the best performing score in each category bolded. The first notable pattern in the results in the superior performance on the helpline data set. This was initially surprising because the enquiry data set had a longer average document length, and the combined data set had the most records. However, the enquiry data still skewed towards shorter documents, and many of the longer documents were the result of advocates including additional technical and logistic language that did not offer much unique or relevant information. Unexpectedly, the vectorization and sentence encoding feature extraction methods performed better than the more advanced deep transfer learning models. Researching this question further would make for interesting future work, but it can be hypothesized that the shorter average

document length and the shorthand note taking style in which the data was collected contributed to the higher performance of more simplistic models.

Table 5: Model Evaluation Metric Comparison.

Data Set	Feature Extraction Method	Optimal Number of Clusters	Silhouette Score	Davis Bouldin Index	Calinski Harabasz Index
Helpline	TF-IDF	10	**0.37**	**1.16**	1143.37
	Doc2Vec	3	0.22	1.40	1359.96
	Distill Roberta	3	0.06	3.80	354.11
	MP-Net	4	0.07	3.45	384.19
Enquiry	TF-IDF	8	0.06	3.60	178.38
	Doc2Vec	3	0.15	2.69	518.46
	Distill Roberta	3	0.05	4.84	327.24
	MP-Net	3	0.05	4.31	242.24
Combined	TF-IDF	12	0.13	2.67	563.65
	Doc2Vec	2	0.23	2.25	**1885.13**
	Distill Roberta	4	0.06	3.87	634.47
	MP-Net	3	0.08	3.55	826.33

The primary factors that hinder performance of vectorization models are high dimensionality and lack of consideration for the semantic ordering of words within a text. The structure of CNNs allows them to resolve such issues, which often accounts for improved performance. However, in this context, most documents were under fifty words and often under twenty, and were recorded as notes summarizing the main points of a conversation. Thus, it is reasonable to assume that entry of notes into Centrepoint's database was more focused on general recall rather than semantic flow. Therefore, the context and features of this data set do not lend themselves to take advantage of the specific benefits the complexity of DNNs are designed to bare. There are numerous studies that prove the superior generalized performance of transformer models and DNNs over vectorization models, but this analysis reiterates that exceptions to the rule always exist, and that it is crucial to consider the specifications of a data set when selecting the model that will most accurately represent its features.

5.2 Cluster Analysis of Best Performing Model

This section examines the variation in size, content, and affiliated categorical and numeric features of the clusters generated by the TF-IDF model. Table 6 shows that the ten final clusters were relativity similar in size, indicating positive performance by the algorithm.

Table 6: Count of Records by Cluster.

Cluster ID	Number of Records
1	761
2	1220
3	1113
4	607
5	599
6	388
7	768
8	473
9	506
10	461

The word clouds in Appendix B display variation in the content of each cluster. Findings show that the words which describe a young person's current housing situation are the primary determinates in cluster differentiation. This is unsurprising, given that a person's housing situation would intuitively be in the foreground of the helpline interactions. This is evidenced by the words that are most prevalent and unique to specific clusters, such as 'sofa', 'surfing', and 'friend' in cluster three, 'rough', 'sleeping', and 'currently' in cluster seven, and 'living', 'home', and 'family' in cluster eight.

To augment the word cloud analysis, cluster IDs were added as a column to the original helpline data table via a join on the entry ID column, to generate a descriptive overview of the categorical and numeric features associated with the records in each cluster. Table 7 displays the modal or mean value, for categorical and numeric features respectively, for several key features in the helpline data set. However, there is not much variation in these values among each cluster, and furthermore the mean and modal values most represented in each feature within the clusters are consistent with the those of the entire data set. This indicates that the clustering algorithm did not

capture variation in demographic background as much as it did for housing situation. Because of the skewedness of many features towards one or two dominant values, the lack of overall variation across categorical features was anticipated.

Table 7: Descriptive Cluster Analysis.

	Segment	Housing Situation	Gender	Region	Main cause of homelessness	At risk of Homelessness due to	Age
0	First	Living with family	Female	North West England	Family breakdown	Family breakdown	22.069829
1	Second	Other	Female	London	Not Known	Family breakdown	23.535685
2	Third	Sofa-surfing	Female	London	Family breakdown	Family breakdown	22.992812
3	Fourth	Sofa-surfing	Female	London	Family breakdown	Family breakdown	22.207578
4	Fifth	Living with family	Female	London	Not Known	Family breakdown	22.205686
5	Sixth	Sofa-surfing	Female	North West England	Family breakdown	Family breakdown	22.368557
6	Seventh	Rough sleeping	Male	London	Family breakdown	Family breakdown	23.510417
7	Eighth	Living with family	Female	North West England	Family breakdown	Family breakdown	21.875000
8	Ninth	Other	Female	London	Not Known	Unknown	23.905138
9	Tenth	Living with family	Female	North West England	Not Known	Family breakdown	22.950108

5.3 Organizational Implications

This project marks Centrepoint's first experimentation with NLP and feature extraction, as well as with clustering with ML. The data team will be able to use the insights gained and lessons learned from this project to propose similar future work for different functions of the organization. In terms of the results of the clustering analysis, the model performance did not meet a high enough standard to induce changes in policy or business strategy on its own. However, the limitations of the helpline data and how it is collected caused the Centrepoint to reflect on ways to gather high quality and more appropriate data for similar future analyses. For example, Centrepoint conducts focus groups with young people with whom they have an established relationship with. Data collection from those sessions could potentially produce more robust data than the notes collected at first contact and during a crisis.

Another potential avenue for future analyses is building similar NLP models on the full-length recorded transcripts of the conversations captured in the helpline data. Currently, the representative logs brief notes about the call, which make up the free text features in the data sets accessed in this project. However, Centrepoint does record and store the full transcripts of the conversations, but

currently do not have an efficient method of downloading and storing the contents of the recordings in a structured data set. This limitation prevents a current analysis of the transcript. Modifying the data collection pipeline to make such a project possible could potentially produce more robust results if the recordings contained details excluded from the helpline database.

5.4 Limitations

Several key limitations affected the methodology and results of this analysis. Despite the numerous amounts of features in the original data sets, the extremely high prevalence of null values rendered most of them unusable for ML. In addition, the inconsistencies in the number and type of features in both data sets meant that they could not be wholly combined and treated as a singular source. These facts led the analysis to focus solely on extracting features from the free text features for clustering models. Another limitation was the lack of numeric features, as the k-means model can combine free and text numeric features for unsupervised clustering but cannot accommodate categorical features. If more of the non-text features were of a higher quality or numeric in nature, the analysis could have employed a methodology that included a richer data set, which may have resulted in more distinct clusters.

After the methodology was defined, the inconsistencies in how the text data was collected between data sets and the overall brevity of the document length were factors in diminishing the overall performance of the models. If the entries were longer and contained more varied information, or if the enquiry data set contained less logistical and technical language, perhaps the extracted features could have led to more distinct clusters. In addition, the data was collected from young people and advocates that reached out to Centrepoint by their own volition, which biases the data to those actively seeking help and thus any results cannot be generalized to the general population of young people experiencing homelessness in the United Kingdom. Finally, as evidence by the modal values of the categorical features, this was a top-heavy data set, in that one or two dominant values constituted disproportionate shares of the values of each feature. This consistency may have contributed to the overall below average performance of all the models tested.

6. Conclusions and Future Work

The goal of this dissertation was to evaluate the performance of NLP feature extraction algorithms on unsupervised clustering tasks. The analysis found that the TF-IDF extraction framework produced the greatest relative performance of the algorithms tested, as measured by established

cluster purity criteria. An analysis of the ten clusters the TF-IDF model produced revealed that much of the inter-cluster disparities can be attributed to variation in housing situations described in helpline summary notes. The fact that vectorization models outperformed Deep Learning approaches leveraging CNNs, despite the documented empirical advantages of the latter group, indicates that the traits of this data set did not present the dimensionality nor semantic ordering problems CNN approaches were designed to address in NLP analyses. Thus, the crucial insights of this work concern the importance of considering the traits and structure of a data set in selecting the most appropriate feature extraction framework for a ML task and not solely relying on the documented generalized performance of existing models.

There were several key limitations that arose during the project, however these obstacles prompted numerous proposals for future work. For Centrepoint and their future data projects, conducting similar analyses on data sets with more complete feature sets or extracting features from the unedited transcripts of the helpline interactions may led to more robust and instructive results. From a broader academic perspective, promising continuations in the NLP space may include a comparable analysis using data with lengthier documents, evaluating recently developed feature extraction models that optimize performance on short-text clustering tasks, and employing k-means clustering on data sets with both free text and numeric input features to capture a more complete information set. These are precedented avenues in the public health data field and would expounding upon them with cutting edge data science methodologies would make for exciting extensions of this analysis.

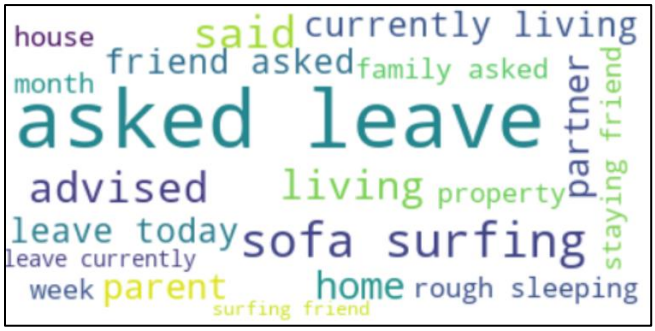
Appendices

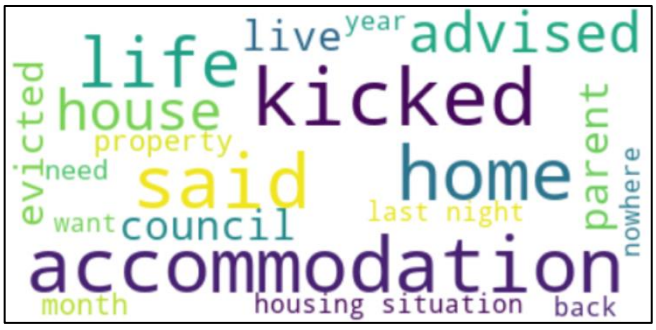
Appendix A- Links to Project Code and Resource Board

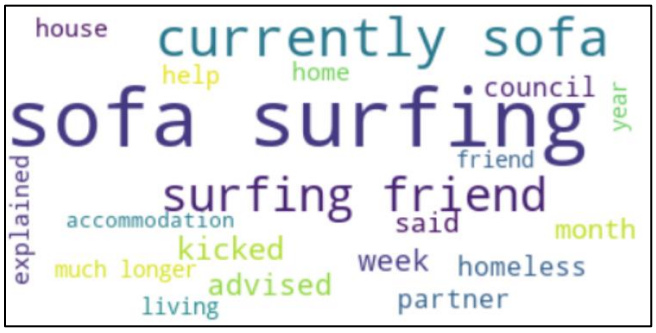
The code for this dissertation can be found on the following GitHub repository: https://github.com/geacosta94/Masters-Dissertation.

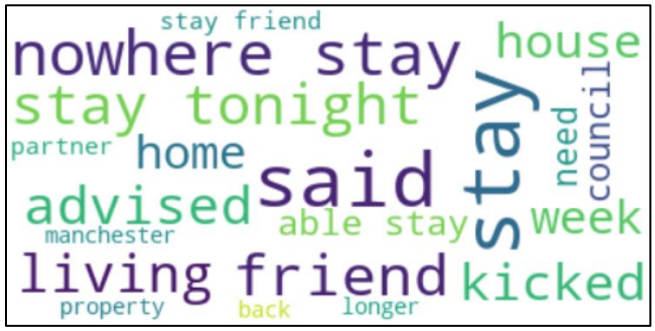
The Trello board tracking the progress of this dissertation can be found here: https://trello.com/b/IyL65hXQ/disseration-project-management-board.

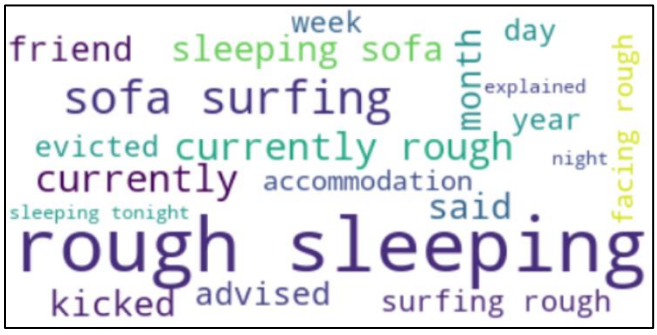
Appendix B- Cluster Word Clouds for TF-IDF Model

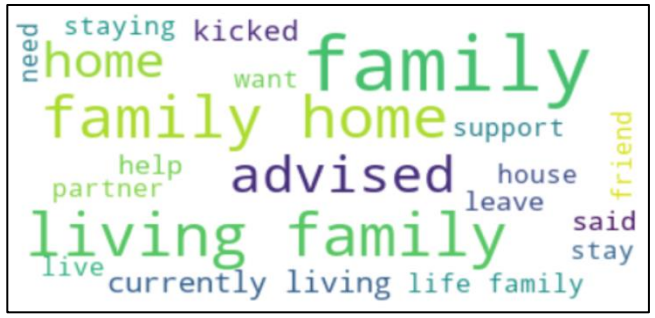


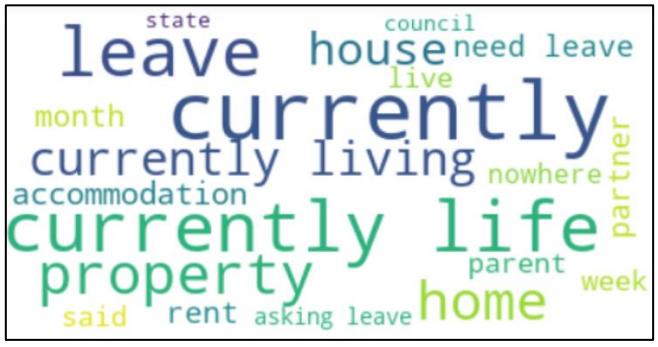












Bibliography

Aggarwal, C.C. and Zhai, C. (2012). A Survey of Text Clustering Algorithms. *Mining Text Data*, [online] pp.77–128. Available at: https://link.springer.com/chapter/10.1007/978-1-4614-3223-4_4 [Accessed 6 May 2021].

Arnaud, E., Elbattah, M., Gignon, M. and Dequen, G. (2020). Deep Learning to Predict Hospitalization at Triage: Integration of Structured Data and Unstructured Text. *In Proceedings of the 2020 IEEE International Conference on Big Data (Big Data)*. [online] Available at: https://ieeexplore.ieee.org/abstract/document/9378073 [Accessed 6 May 2021].

Arora, M. and Kansal, V. (2019). Character level embedding with deep convolutional neural network for text normalization of unstructured data for Twitter sentiment analysis. *Social Network Analysis and Mining*, 9(1).

Arthur, D. and Vassilvitskii, S. (2007). K-Means++: The Advantages of Careful Seeding. In: *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. [online] SODA 2007. DBLP. Available at: https://www.researchgate.net/publication/220778887_K-Means_The_Advantages_of_Careful_Seeding [Accessed 11 Jul. 2021].

Ben Salem, S., Naouali, S. and Chtourou, Z. (2018). A fast and effective partitional clustering algorithm for large categorical datasets using a k -means based approach. *Computers & Electrical Engineering*, 68, pp.463–483.

Bird, Steven, Loper, E. and Klein, E. (2009). *Natural Language Processing with Python*. O'Reilly Media Inc.

Bojanowski, P., Joulin, A., Grave, E. and Mikolov, T. (2016). Enriching Word Vectors with Subword Information.

Cannon, R.L., Dave, J.V. and Bezdek, J.C. (1986). Efficient Implementation of the Fuzzy c-Means Clustering Algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(2), pp.248–255.

Chen, R., Ho, J.C. and Lin, J.-M.S. (2020). Extracting medication information from unstructured public health data: a demonstration on data from population-based and tertiary-based samples. *BMC Medical Research Methodology*, [online] 20(1). Available at: https://link.springer.com/article/10.1186/s12874-020-01131-7; [Accessed 26 Jul. 2021].

Christian, H., Agus, M.P. and Suhartono, D. (2016). Single Document Automatic Text Summarization using Term Frequency-Inverse Document Frequency (TF-IDF). *ComTech: Computer, Mathematics and Engineering Applications*, [online] 7(4), p.285. Available at: https://journal.binus.ac.id/index.php/comtech/article/view/3746 [Accessed 3 Aug. 2019].

Dangeti, P. (2017). *Elbow Method for Selection of Optimal k Clusters. Statistics for Machine Learning.* Available at: https://learning.oreilly.com/library/view/statistics-for-machine/9781788295758/c71ea970-0f3c-4973-8d3a-b09a7a6553c1.xhtml.

Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *In Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*.

Doermann, D.S. (1998). An introduction to vectorization and segmentation. *Graphics Recognition Algorithms and Systems*, pp.1–8.

Elbattah, M. and Molloy, O. (2017). Data-driven patient segmentation using K-means clustering: The case of hip fracture care in Ireland. *In Proceedings of the Australasian Computer Science Week Multiconference*.

Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J. and Chen, T. (2018). Recent advances in convolutional neural networks. *Pattern Recognition*, [online] 77, pp.354–377. Available at: https://www.sciencedirect.com/science/article/pii/S0031320317304120 [Accessed 23 Jul. 2021].

He, C., Chang, J. and Chen, X. (2010). Using the Triangle Inequality to Accelerate TTSAS Cluster Algorithm. *2010 International Conference on Electrical and Control Engineering*.

Honnibal, M. and Montani, I. (2017). spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing.

Jin, C. and Bai, Q. (2019). Short Text Clustering Algorithm Based on Frequent Closed Word Sets. *In Proceedings of the 2019 12th International Symposium on Computational Intelligence and Design (ISCID)*. [online] Available at: https://ieeexplore.ieee.org/document/9092562/authors#authors [Accessed 8 Jul. 2021].

Jurafsky, D. and Martin, J.H. (2014). *Speech and language processing : an introduction to natural language processing, computational linguistics, and speech recognition*. India: Dorling Kindersley Pvt, Ltd.

Kalchbrenner, N., Grefenstette, E. and Blunsom, P. (2014). *A Convolutional Neural Network for Modelling Sentences*. [online] ACLWeb. Available at: https://www.aclweb.org/anthology/P14-1062/.

Kaushik, M. and Mathur, M. (2014). *Comparative Study of K-Means and Hierarchical Clustering Techniques*. [online] *https://www.researchgate.net/*. International Journal of Software and Hardware Research in Engineering. Available at: https://www.researchgate.net/publication/293061584_Comparative_Study_of_K-Means_and_Hierarchical_Clustering_Techniques [Accessed 6 May 2021].

LeCun, Y., Boser, B.E., Denker, J., Henderson, D., Hubbard, W. and Howard, R. (1989). Handwritten digit recognition with a back-propagation network. In: *In Proceedings of Advances in Neural Information Processing Systems (NIPS)*. pp.396–404.

Lecun, Y., Bottou, L., Bengio, Y. and Haffner, P. (1998). Gradient-based learning applied to document recognition. *In Proceedings of the IEEE*, 86(11), pp.2278–2324.

Lev, G., Klein, B. and Wolf, L. (2015). In Defense of Word Embedding for Generic Text Representation. *Natural Language Processing and Information Systems*, [online] pp.35–50. Available at: https://www.researchgate.net/publication/300786368_In_Defense_of_Word_Embedding_for_Generic_Text_Representation [Accessed 7 Jun. 2021].

Li, R. and Shindo, H. (2015). Distributed Document Representation for Document Classification. *Advances in Knowledge Discovery and Data Mining*, pp.212–225.

Liang, H., Sun, X., Sun, Y. and Gao, Y. (2017). Text feature extraction based on deep learning: a review. *EURASIP Journal on Wireless Communications and Networking*, 2017(1).

Loria, S. (2018). *TextBlob: Simplified Text Processing — TextBlob 0.15.2 documentation*. [online] Readthedocs.io. Available at: https://textblob.readthedocs.io/en/dev/.

Maklin, C. (2019). *Gaussian Mixture Models Clustering Algorithm Explained.* Available at: https://towardsdatascience.com/gaussian-mixture-models-d13a5e915ce8e [Accessed 30 Jul. 2021].

Manning, C.D, Raghavan, P. and Schutze, H. (2009). *Introduction to Information Retrieval.* Cambridge: Cambridge University Press.

McCallum, J. and Sondej, F. (2021). *GitHub - filyp/autocorrect: Spelling corrector in python*. [online] GitHub. Available at: https://github.com/filyp/autocorrect [Accessed 21 Jul. 2021].

Merkx, D. and Frank, S.L. (2021). Human Sentence Processing: Recurrence or Attention? *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*.

Mikolov, T., Corrado, G. and Chen, K. (2013). Efficient Estimation of Word Representations in Vector Space. In: *Conference: Proceedings of the International Conference on Learning Representations (ICLR 2013)*.

Mikolov, T., Kombrink, S., Burget, L., Černocký, J. and Khudanpur, S. (2011). *Extensions of recurrent neural network language model*. [online] IEEE Xplore. Available at: https://ieeexplore.ieee.org/document/5947611.

Mikolov, T. and Le, Q. (2014). Distributed Representations of Sentences and Documents. *Google*. [online] Available at: https://arxiv.org/abs/1405.4053 [Accessed 11 Jul. 2021].

Nguyen, T.H. and Grishman, R. (2015). Relation Extraction: Perspective from Convolutional Neural Networks. *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*.

Parmar, R. (2018). *Deep Network Architecture with Multiple Layers*. *Training Deep Neural Networks.* Available at: https://towardsdatascience.com/training-deep-neural-networks-9fdb1964b964 [Accessed 23 Jul. 2021].

Patel, V. and Mehta, R. (2011). Impact of Outlier Removal and Normalization Approach in Modified k-Means Clustering Algorithm. *IJCSI International Journal of Computer Science Issues*, [online] 8(2), pp.331–336. Available at: https://www.researchgate.net/publication/266225875_Impact_of_Outlier_Removal_and_Normalization_Approach_in_Modified_k-Means_Clustering_Algorithm [Accessed 6 May 2021].

Pearlmutter, B. (1989). Learning state space trajectories in recurrent neural networks. *Neural Computation*, 1(2), pp.263–269.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., and et al. (2011). Scikit-learn: Machine Learning in Python. *JMLR*, 12(85), p.2825−2830.

Pennington, J., Socher, R. and Manning, C. (2014). Glove: Global Vectors for Word Representation. *In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. [online] Available at: https://www.aclweb.org/anthology/D14-1162/.

Phung and Rhee (2019). A High-Accuracy Model Average Ensemble of Convolutional Neural Networks for Classification of Cloud Image Patches on Small Datasets. *Applied Sciences*, 9(21), p.4500.

Ram, A., Jalal, S., Jalal, A.S. and Kumar, M. (2010). A Density Based Algorithm for Discovering Density Varied Clusters in Large Spatial Databases. *International Journal of Computer Applications*, 3(6), pp.1–4.

Rashid Ahmed Ahmed, S., Al Barazanchi, I., Jaaz, Z.A. and Abdulshaheed, H.R. (2019). Clustering algorithms subjected to K-mean and gaussian mixture model on multidimensional data set. *Periodicals of Engineering and Natural Sciences (PEN)*, 7(2), p.448.

Rehurek, R. and Sojka, P. (2011). Gensim–python framework for vector space modelling. In: *NLP Centre*. Brno, Czech Republic: Masaryk University.

Reimers, N. (2021). *Pretrained Models — Sentence-Transformers documentation*. [online] www.sbert.net. Available at: https://www.sbert.net/docs/pretrained_models.html.

Roell, J. (2017). *Neural Networks That Cling to the Past*. *Understanding Recurrent Neural Networks: The Preferred Neural Network for Time-Series Data*. Available at: https://towardsdatascience.com/understanding-recurrent-neural-networks-the-prefered-neural-network-for-time-series-data-7d856c21b759 [Accessed 23 Jul. 2021].

Sarwan, N. (2017). *Count Vector Representational Image*. *An Intuitive Understanding of Word Embeddings: From Count Vectors to Word2Vec*. Available at: https://www.analyticsvidhya.com/blog/2017/06/word-embeddings-count-word2veec/ [Accessed 23 Jul. 2021].

Sculley, D. (2010). Web-scale k-means clustering. *In Proceedings of the 19th international conference on World wide*

web - WWW '10.

Stevens, E., Dixon, D.R., Novack, M.N., Granpeesheh, D., Smith, T. and Linstead, E. (2019). Identification and analysis of behavioral phenotypes in autism spectrum disorder via unsupervised machine learning. *International Journal of Medical Informatics*, 129, pp.29–36.

Stoyanov, V., Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M. and Zettlemoyer, L. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. In: . Facebook.

Tiedemann, J. and Lison, P. (2016). OpenSubtitles2016: Extracting Large Parallel Corpora from Movie and TV Subtitles. In: *In Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*.

Vaswani, A., Shazeer, N., Jones, L., Parmer, N., Uszkoreit, J., Kaiser, Ł., Gomez, A. and Polosukhin, I. (2017). Attention Is All You Need. *In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017)*.

Vogler, R. (2014). *TF-IDF Derivation. The tf-idf-Statistic For Keyword Extraction.* Available at: https://www.r-bloggers.com/2014/02/the-tf-idf-statistic-for-keyowrd-extraction/ [Accessed 25 Jul. 2021].

Waldron, R., O'Donoghue-Hynes, B. and Redmond, D. (2019). Emergency homeless shelter use in the Dublin region 2012–2016: Utilizing a cluster analysis of administrative data. *Cities*, [online] 94, pp.143–152. Available at: https://www.sciencedirect.com/science/article/pii/S0264275118314045 [Accessed 8 May 2021].

Wang, P., Xu, B., Xu, J., Tian, G., Liu, C.-L. and Hao, H. (2016). Semantic expansion using word embedding clustering and convolutional neural network for improving short text classification. *Neurocomputing*, 174, pp.806–814.

Xu, J., Xu, B., Wang, P., Zheng, S., Tian, G., Zhao, J. and Xu, B. (2017). Self-Taught convolutional neural networks for short text clustering. *Neural Networks*, [online] 88, pp.22–31. Available at: https://www.sciencedirect.com/science/article/pii/S0893608016301976 [Accessed 20 Feb. 2020].